A MULTIMEDIA-BASED BILINGUAL INSTRUCTIONAL SYSTEM USING AN EXPERT SYSTEM SHELL

by

Tung-Ying Chang

An Abstract

of a thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Department of Library Science and Information Services
Central Missouri State University

April 1992

Thesis Chair: Professor Aileen Helmick

ABSTRACT

The objective of this research was to create a programming-free tutoring system which will enable teachers to generate low-cost multimedia-based bilingual courseware for a variety of subjects.

The study was based on the contents of three main disciplines: linguistics, artificial intelligence, and knowledge engineering. The bilingual expert system currently runs on IBM PC compatible computers and includes a knowledge base, an original program to create and manage text files, and sample courseware to test the system. Both expert system techniques and conventional programming were used to construct the system.

The following concepts were implemented in the development of this system: course-independent design, graphics to present Chinese characters, the combination of a database structure and word processing techniques to build the text file editor, and use of the "hungry-tracing method" in the hybrid inference of forward chaining and backward chaining.

The design procedure is also discussed in the thesis.

Abstract approved: _____, Thesis Chair

_____, Date

A MULTIMEDIA-BASED BILINGUAL INSTRUCTIONAL SYSTEM USING AN EXPERT SYSTEM SHELL

A Thesis

Presented to

the Faculty of the Department of Library Science and Information Services

Central Missouri State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

---

by

Tung-Ying Chang

April 1992

A MULTIMEDIA-BASED BILINGUAL INSTRUCTIONAL SYSTEM USING AN EXPERT SYSTEM SHELL

by

Tung-Ying Chang

Has been Approved for the Department
of
Library Science and Information Services

CENTRAL MISSOURI STATE UNIVERSITY
WARRENSBURG, MISSOURI

By

Thesis Chair

Committee Member

Committee Member

April 1992

A MULTIMEDIA-BASED BILINGUAL INSTRUCTIONAL SYSTEM USING AN EXPERT SYSTEM SHELL

by

Tung-Ying Chang

Has been Approved for the Department
of
Library Science and Information Services

CENTRAL MISSOURI STATE UNIVERSITY
WARRENSBURG, MISSOURI

By

_____
Thesis Chair

_____
Committee Member

_____
Committee Member

April 1992

ACKNOWLEDGEMENTS

I wish to acknowledge the following people for their support during the preparation of this research study. Through Central Missouri State University, I want to thank my thesis advisor, Dr. Aileen Helmick who has supervised my work on this thesis. She has been beneficial in aiding me throughout my academic research and in developing my ideas. Her patience and enthusiasm have encouraged me while writing this paper. I also thank Professor Philip Sadler and Dr. Charles E. Slattery, members of my thesis committee, for reviewing the manuscript and offering many suggestions. I am especially grateful to Dr. Ronald W. Long, my former professor of computers in linguistics, for his instruction, support, and encouragement. Through Cheng Kung University, I want to express my sincere appreciation to professor Ching-Hsiung Lee, Dr. Hong-Sen Yun, Dr. Charles Lewis, and Dr. Sue-J. Lin, whose encouragement and vision help me to adapt myself to the world of artificial intelligence. Through Taipei Institute of Technology, I appreciate my former professor Su-Chin Pan, for her instruction and encouragement.

This research study is dedicated to my mother Cheng-Chu and my wife Su-Hsing. Without their endless love and unconditional support, I would never have the opportunity to enter the palace of knowledge and fulfil my dream.

1992

Tung-Ying Chang

ALL RIGHTS RESERVED

TABLE OF CONTENTS

CHAPTER I INTRODUCTION Page

Background 1

Statement of the Problem 2

Purpose of the Study 2

Limitations of the Study 3

Definition of Terms 3

Significance of the Study 5

Organization of the Study 5

CHAPTER II REVIEW OF RELATED LITERATURE

Introduction 7

Linguistics and Intelligence 7

Production System 11

Artificial Intelligence 13

Knowledge Engineering 17

Knowledge-based System 17

Expert System 18

Expert System Development Stages 19

Computer-Assisted Instruction 19

Interactive Multimedia 21

Intelligent Tutoring System 23

Summary of the Literature 25

CHAPTER III METHODOLOGY

Basic Design Concepts 27

Selection of an IBM PC Compatible Computer for Software Development and User Environment 27

Consideration for Developing the System by Using an Expert System

Development Shell 28
Choice of a High-level Programming Language to Write Auxiliary Programs 28
Course-independent Design 29
Employment of Rule-based Representation 29
Use of Hierarchical Fames for Organizing Basic Structure of the System 29
Top-down Design 30
Design of External Software Access Interface to Incorporate with Application Software 30
Creation of Tutorial Modules to Perform Instruction Strategies 30
Use of Graphics to Present Chinese Characters 30
Design Procedure 31
Resources Assessment and Tools Selection 32
Human resources 32
Shell selection 32
Auxiliary programming language selection 34
Software needed 34
Hardware needed 36
Pilot Test 36
Sketch Operating Flow Diagram 37
Building Knowledge Base 38
Editing Tool Design 39
Courseware Design 39
System Testing 40
CHAPTER IV SYSTEM ANATOMY
System Overview 41
Operating Procedure 41
Operating Flow Diagram 50
General Architecture of the System 53
Programming Tactics 55

Frame Instantiation 55
External Program Interface 56
Chinese Characters and Graphics Processing 57
Text File Editor 58
CHAPTER V CONCLUSIONS AND RECOMMENDATIONS
Conclusions 61
Recommendations for Future Research 63
REFERENCES 67

LIST OF FIGURES

Figure		Page
2-1.	Picture for intelligence test	10
2-2.	Six-year old boy's creative thinking through the representation of graphics ...	13
3-1.	Frame structure and properties of Personal Consultant Plus	33
3-2.	Features analysis of Personal Consultant Plus	35
3-3.	Graphic input function of the first pilot system	37
4-1.	First title page of TITES	42
4-2.	Second title page of TITES	42
4-3.	User information request form	43
4-4.	Functional organization diagram of TITES	43
4-5.	Contents screen in the tutor module	44
4-6.	Multiple choice question of Test-A	46
4-7.	Score report of the test result	46
4-8.	Screen showing a question, user's answer, and correct answer	47
4-9.	Instruction menu of Test-A	47
4-10.	Instruction screen of Test-A	48
4-11.	Graphic-based multiple choice questions of Test-B	48
4-12.	Test-C: Ten blanks for users to complete	49
4-13.	Questions, user's answers, and correct answers for Test-C	49
4-14.	Reading comprehension test of Test-E	50
4-15.	Final score report of the examination	50
4-16.	Typical operating flow diagram of TITES	51
4-17.	Overall frame-tree architecture of TITES	54
4-18.	Main menu of the Text File Editor	59
5-1.	Development Time in Hours: Where the Time Went	62
5-2.	Digital sound format of a Taiwanese folk song presented in the MacRecorder	65

CHAPTER I

INTRODUCTION

An animal watches it,
a human observes it,
a philosopher thinks it,
a scientist studies it,
an engineer builds it.

Tung-Ying Chang

Background

The 1990s is an exciting age when microcomputers are available to most members of industrial societies, schools, and families. The computer allows processes to be accomplished faster, more reliably, and with less human effort. In 1987, many expert system building tools, or "shells," were introduced; some of them are sophisticated and inexpensive. These tools not only make expert system technology available to personal computer users but allow applications to be built in less time than with Artificial Intelligent (AI) languages.

In the summer of 1985, the author was interested in the scope of natural language translation systems and was unable to make a breakthrough in this area (Chang, 1985). In 1987, the use of CAI in Taiwan schools was in its infancy. At the same time, in carrying out its basic purpose of developing authoring systems to facilitate CAI in the Chinese language, a six-year plan was designed by the government of Taiwan to support CAI education. The goals of the plan were to develop CAI, to popularize computer concepts, and to provide effective and efficient instruction in education (Alessi & Shih, 1989).

During the period of 1986-1990, research of CAI in Taiwan concentrated on Chinese CAI authoring systems and the evaluation of using CAI in education, however, few reports revealed that intelligent CAI programs were developed with expert system technology. By the end of 1987, after reviewing the related literature and writing two small pilot expert systems, the author was engaged in a major effort aimed at developing a multimedia-based bilingual instructional system using an expert system shell.

Statement of the Problem

Early CAI experiments in Taiwan were undertaken on mainframe computers but had not been considered successful (Wu, 1987). The main reasons, perhaps, are high installation costs and difficulties of use. In general, mainframe computers are far more expensive than microcomputers. In the conventional CAI system, the teacher uses an authoring language to develop courseware. Some sophisticated authoring systems are powerful enough but somewhat more difficult for novices (Dever & Pennington, 1989). Another difficulty is the Chinese language itself. The Chinese language is not based upon an alphabetic system; thousands of different hieroglyphic characters are included in current Chinese language. CAI programs for Chinese students must be bilingual with the ability of graphics processing. Chinese instruction is necessary for elementary and secondary school students in Taiwan.

Although Chinese input systems had been developed by 1985 or earlier, the interface problems between application software and Chinese input systems still exist. Chinese input systems take more memory space than English systems. Character input and processing slow down response time.

In Taiwan, traditional CAI systems are limited to special curricula such as English and mathematics and result in higher software design cost. Most CAI software lack CMI functions. In addition, instructional methodologies are not varied, consisting mostly of tutorial, drill, and instructional game programs. Few development efforts are based on a rigorous instructional design model (Alessi & Shih, 1989).

Purpose of the Study

One of the initial goals of the study was to develop a concrete understanding of CAI and expert system techniques through a practical design process. The ultimate objective was to create a programming-free instructional system which will enable teachers to generate low-cost multimedia-based bilingual courseware for a variety of subjects.

The study should promote the whole system as a means to integrate CAI, database, expert system, and multimedia technologies in education.

Limitations of the Study

During the last decade, there have been endless changes in the world of computers, including both hardware and software. The progress of microcomputers, mass storage media, and image and sound processing technologies was beyond imagination in earlier years, however, many of them are quite expensive and take time to learn to use. The success of a CAI system will often lie in the effectiveness and friendliness of the user environment. No matter how powerful the system or how sophisticated its design, if users cannot afford the expense, then the system will not achieve its goal; therefore, the study is limited in selected hardware and software. More specifically, instead of trying to combine expensive facilities, the system was developed for the MSDOS Version 3.30 operating system with IBM compatible microcomputers and is based upon a run-time version of an expert system development shell--Personal Consultant Plus Version 3.0.

Although digital sound effects can be adapted to the system, the production costs and related equipment of audio media are relatively high to users. For cost-effectiveness, only text, graphics, and animation are used as instructional media in the current system. Another limitation is courseware design. As it was not included in the study, teaching and testing materials used for system testing have not been validated.

Definitions of Terms

The following terms and definitions will be used in this paper:

Algorithm -- A fixed programming procedure designed for a specific function, or a set of instructions for solving a problem.

Application software -- A set of design programs that allows users to perform specific tasks. Examples include word processors, data management programs, and CAI software.

AI -- Artificial Intelligence, a term coined by John McCarthy in the mid-1950s. AI is a field concerned with designing computer systems that can mimic human intelligence.

AI language -- A programming language popular in the field of Artificial Intelligence such as LISP and PROLOG.

Authoring language -- A high-level programming language designed specifically for creating CAI or educational programs.

CAI -- Computer-assisted instruction. A generic term that includes a wide range of types of computer programs for instructional purposes.

CD-ROM -- Compact disc-read only memory. Discs principally used for massive data storage.

Digital CD -- Includes audio CD, videodisc, and CD-ROM.

Chinese input system -- Computer software designed for using a keyboard to input Chinese characters into a microcomputer.

CMI -- Computer-managed instruction, a program concerned with record keeping, test grading, and data management of classroom instructional activities.

Consultation -- Consultation refers to user interaction with an expert system in a computer.

Expert system building tool -- Computer software or design tool that facilitates the development of expert systems. These tools are built upon programming languages.

Frame structure -- Frame refers to a special way of representing common concepts and situations. It was introduced by Marvin Minsky in 1975. It arises from the realization that many objects, acts, and events are "stereotyped" (Minsky, 1975).

Graphics -- Anything that appears on the screen.

Hardware -- A general term for physical devices for computers, such as monitors, system units, scanners, printers, keyboards, and disk drivers. The term is often used in opposition to software.

Instantiation -- Instantiation refers to the process by which PC Plus activates or enters a frame during a consultation. When PC Plus instantiates the frame during a consultation, the frame is given a dynamic, concrete reality (Texas Instruments, 1987).

Mainframe -- A large, time-sharing computer that allows many users to work with different application software simultaneously.

Microcomputer -- Synonymous with personal computer, a computer in the lower price range which uses a microprocessor as CPU.

Shell -- See Expert system building tool.

Significance of the Study

A human tutor can be costly and affordable only to a wealthy student. This study may offer a practical tool for teachers to generate low-cost PC-based CAI courseware for the purpose of education or professional training. The benefits could be extended to researchers or educators who are interested in the field of students' learning model studies, instructional technology, and curriculum design.

Furthermore, this approach may reveal that an expert system which becomes more intelligent will be practical for educational applications, not just as a research tool. Because of expert system development shells, expert systems no longer require many work-years to develop as did earlier expert systems in the 1970s. In the area of AI research, not only research institutions, but individual researchers can engage in high-tech study independently.

Organization of the Study

Chapter I introduces the study, provides its background, states the problems, and describes its purpose, limitations, and significance. The ultimate objective of the study is to create an intelligent tutoring system, and in order to understand this endeavor, it will be necessary to examine both the theory of linguistics and the experimental investigations of artificial intelligence. In chapter II, the literature related to these topics will be reviewed; begin with discussing the relationship between linguistics and intelligence, AI, knowledge engineering, CAI, interactive multimedia and intelligent tutoring system will be explored systematically.

Chapter III will describe the author’s research method; basic design concepts and the design procedure. In chapter IV, the operating procedure, flow diagram, general architecture, and programming tactics of the system will be discussed. The concepts and ideas implemented in the development of the system will also be mentioned.

Finally, in chapter V, conclusions and recommendations of future research will be presented.

CHAPTER II

REVIEW OF RELATED LITERATURE

All that glisters is not gold --

William Shakespeare

Introduction

The study was based on three main scientific fields: linguistics, artificial intelligence, and knowledge engineering that were founded and developed by AI pioneers and researchers since the 1950s. Many ideas and concepts were borrowed from disciplines including linguistics, cognitive psychology, artificial intelligence, computer-assisted instruction, and software engineering.

Linguistics and Intelligence

Linguistics is more than a simple branch of social science and probably is the most exact of all. It can be used to predict what is going on in this human world, and prediction can be the ultimate test of any science. Linguists believed that language is not only a tool for communication but also a form of thought. It is the very substance that constitutes ideas and cultures. Indo-European languages are based upon dichotomies or two-valued logic (e.g., good and evil), subject-predicate structure (e.g., The wind is blowing.) and the law of identity (e.g., A is A. "This is bread." The "to be" structure in English is responsible for a vast number of ideas conforming to the law of identity). Two-valued logic and the law of identity have dominated Western thinking for thousands of years. As Benjamin L. Whorf pointed out, the form of a man's thoughts is controlled by patterns learned early, of which the man is mostly unconscious. Thinking is the process of language in operation (Chase, 1954).

How does a given language mold the thoughts of humans and build their views of nature and the world? Like the question of what constructs the nature of mind, it is not unusual that these issues cause frustration. Yet, research in this area promises an encouraging future. Metalinguistics aims at examining the application of language systems and language response as well as the impact of language on human thoughts and acts,

which provides an area of further studies for psycholinguistics, comparative linguistics, cognitive psychology, and sociolinguistics.

There is general consensus that language is a tool of thinking. If it can be assumed that human thinking relies on some common logic rules, there must be one single rhetorical structure for all human beings. In fact, different languages are related to different thought patterns. The relativity of language leads to various views of nature and the universe and cultural differences as well. Nevertheless, it is not surprising to find out upon further analysis that different cultural patterns still share some important similarities despite this overwhelming variety (Slobin, 1979).

In order to understand the relation between language and intelligence, it is necessary to retrospect the views of Chomsky and Piaget. Chomsky believed that language is an inane mechanism of human beings (Chomsky, 1980). Piaget, in contrast, by adopting the experimental results of psychologists, argues that language ability can be improved by raising the intelligence level, whose improvement, nevertheless, has nothing to do with language ability (Slobin).

From the biological point of view, intelligence is the nature of biological mechanism and one of the characteristics of biological heritage. Except for the lack of motor cortex and language ability, advanced animals possess a similar nerve system to that of human beings. Although some insects are able to pass on messages through certain signals, there is yet insufficient evidence to prove whether such an experience is just inherited from generation to generation through genes. Monkeys, on the other hand, undergo the same process of growing and maturing as human beings by gathering new experiences. Evidence shows that a baby monkey deals with a complicated problem more effectively than a human baby of the same age (Chase). Language makes it possible for human knowledge to grow accumulatively; therefore, there is no comparison between the two. Human intelligence is unsurmountable; however, there is one animal in this world that probably is more intelligent than human beings, the dolphin. It has a larger than human brain plus a highly advanced organization of motor cortex that enables it to memorize words. According to zoologists, the super sound wave cries emitted by the dolphin represent a type of language.

Consequently, experts are able to communicate with dolphins by using certain language skills. The above facts illustrate the unusual relationship between language and intelligence. In the past, researchers resorted to the biological instincts theory whenever animals displayed evidence of intelligence. The time has come for further studies to be conducted from theoretical perspectives of physics and chemistry besides biology.

In 1933, Bloomfield pointed out, in his famous study of Jack and Jill, that language is a stimulus-response phenomenon (Palmer, 1981). He presented the following example: The boy, Jack, was travelling with the girl, Jill. One day, Jill was hungry. She saw an apple on the tree. She would have picked the apple herself had Jack not been present. This is a typical example of stimulus (hunger) and response (picking the apple) phenomena. Since Jack was with her, the stimulus did not cause immediate response but, through language response, she told Jack she was hungry. The sound wave reached Jack as a language stimulus and caused Jack to pick the apple. The whole process is illustrated as follows: S -- [R] ...[S] -- R. The point is that both stimulus and response are physical phenomena.

Another example is the biological response of Escherichia coli, an intestinal bacterium. The germ processes information about its chemical environment, sensing twenty different substances at a time. It is observed that the individual germs swims not just in the direction of the nutrient, but toward where the nutrient is increasing at the fastest rate. Every four seconds, it reevaluates the information of its environment (Lemonick, 1984).

From the above scientific phenomenon, it seems that Aristotle's two-valued syllogisms also exist in germs. Some people might argue that germs move by biological instincts or that a germ is a simple stimulus-response organism. If we compare this with Bloomfield's argument, isn't it possible that the germ's instincts or its single-cell intelligence carries a pattern similar to that of human language? Can scientists presume that natural language is a reflection of the innate biological language?

Single-cell intelligence does not provide sufficient evidence to support Chomsky's innateness of human language. Grace de Laguna emphasized that most animals are only able to pass on messages. They cannot think because thoughts which are not formulated are

something less than thought. Real thinking has to be expressed by appropriate means. Even though we believe that an animal cannot express its thought in language because it has no thoughts to express, it is still difficult to infer that all living things have language or similar to language intelligence (Chase).

Piaget did not disagree with Chomsky's theory of innateness. He simply tried to emphasize that the relation between natural language and intelligence is not only an issue of biology. There are actually multi-influences on the development of language and intelligence. The following experiment between a mother and her child is an example (Chang, 1987a).

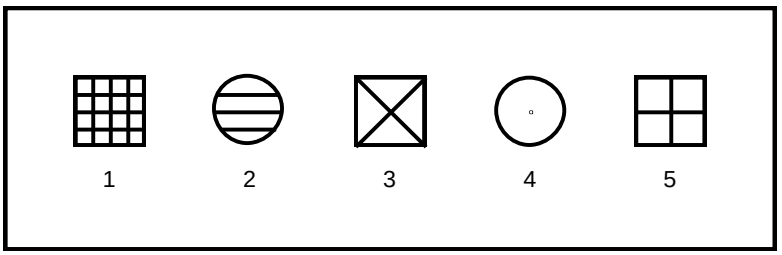


Figure 2-1. Picture for intelligence test.

The boy was shown the five pictures in Figure 2-1 and asked to point out the one that is different from the others. Pictures 1, 3, 4, and 5 are symmetrical; when rotated 90 degrees, these pictures do not change their form. It took him just a little while to pick out the second picture and correctly explain the reason. It took his mother 20 seconds to figure out the answer. The boy was only six, with no formal training and no distinctive talents. The mother had a college degree in English and ten years of work experience. The result shows that the mother with a higher language proficiency is not necessarily superior in intelligence to the boy who is just beginning to learn language.

Other experiments like the IQ test have produced similar conclusions. In Taiwanese, sayings like, "He's even more stupid than a child" or "A clever baby does not mean he'll be an intelligent adult" can be understood. Is it true that the boy has a higher IQ than his mother? The answer is "no." Like Chomsky and Piaget, most people have ignored the social influence on human intelligence. If one starts to observe and study language as a social phenomenon, it is not difficult to discover the strong sociocultural impact on language and

intelligence. Why did it take 20 seconds for the mother to find the answer? Why is mathematics the most precise language? Why are modern physicists trying to open their minds to prevent limitations of their mother tongues? The reasons are not inscrutable.

Production System

Is it possible to understand the mystery of intelligence after studying the relation between language and intelligence from the perspectives of linguistics, biology, psychology, sociology, etc.? The answer is not yet encouragingly positive. Despite the incredible progress of science, human beings still have a very limited understanding of the functions and operations of the human brain. The Nobel Prize winner Professor Herbert Simon, who teaches psychology and computer science at Carnegie-Mellon University, believes the only way to solve the riddle of intelligence and cognition is by using external simulation unless significant progress can be made in brain physiological anatomy and biological science. In 1956, Allen Newell, J. C. Shaw, and Herbert Simon developed the “Logic Theorist” (Newell, Shaw, & Simon, 1957). Their purpose in working on artificial intelligence was to simulate the problem-solving operations of the human mind rather than to make the computer think smart. Their most distinguished contribution in the area of AI is the development of a production system.

What is a production system? It is the simulation of recognition processes. According to Newell and Simon (1972), the recognition process is made up of basic units or conditional statements. A person looks at the sky and says "It's going to rain" because he has done the following thinking: if the sky is dark, then it's going to rain.

The intelligence activity occurs when a certain premise leads to a certain conclusion or a certain situation causes certain actions. A couple of basic intelligence activities combine to form the complicated cognitive process and accomplish full understanding of the facts. This is the fundamental theory of a production system, which also puts an emphasis on the following ideas:

First, structure is far more important than individual facts. There is a general consensus among scientists at this point. Chomsky considered linguistic abilities to be

based on mental structures of rules while Piaget believed "there is no structure apart from construction, either abstract or genetic" (Piaget, 1970, p. 140).

Second, the learning process of human beings is characterized by creativity. Figure 2-2 shows how the boy is doing creative thinking through the representation of graphics. The graphics without the numbers were drawn by the boy. A close study reveals the fact that children usually create vocabulary through inferences. When a child learns the concept of "full of water" and its underlying language structure in Taiwanese, he starts to create expressions like "full of sleep" or "full of food" to describe basic needs (Chang, 1987a).

Third, feedback is a response activity. Man is an organic whole that is capable of giving feedback. A normal person stands on his feet as a result of feedback. It is reasonable to conclude that feedback is a way of control in which each step is modified by its previous step. The same process goes on in the simulation system when input is controlled by output feedback. The often mentioned heuristic decision is another type of feedback.

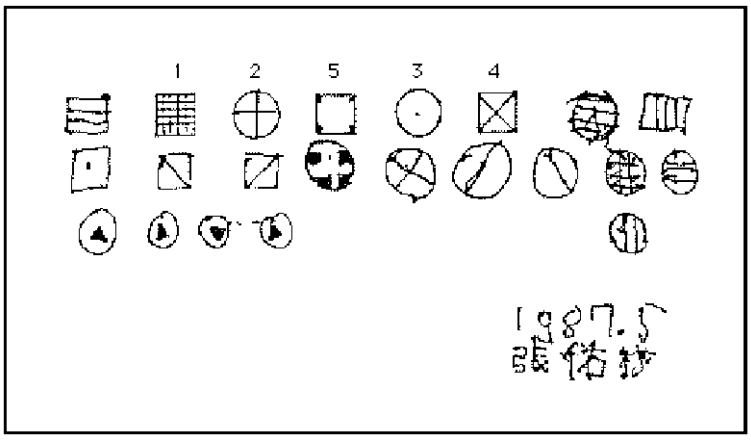


Figure 2-2. Six-year old boy's creative thinking through the representation of graphics.

Due to the complexity of human cognitive processes caused by the necessity of providing appropriate feedback in a variety of situations, the cognitive simulation system has to include a flexible control structure to manipulate facts and rules so that the system does not stall. What makes this system possible? Is there any relation between the production system theory and the physiological, psychological, and sociological bases of language intelligence? Simon and Newell's theory bears a close resemblance to Chomsky's idea of innateness in that they both advocate that the organization of human intelligence operation is

very similar to production structures, an idea which comes more or less from their scientific intuitions. Nevertheless, after more than a decade of experiments, the production system has proved to be not only theoretically valid but also empirically valuable.

Artificial Intelligence

> The AI handbook provided the following definition for artificial intelligence:
>
> Artificial Intelligence is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior -- understanding language, learning, reasoning, solving problems, and so on (Barr & Feigenbaum, 1981, Vol. I, p. 3).

Before World War II, formal logic and cognitive psychology were considered scientific fields of study. Vannevar Bush (1945) presented a hypermedia-like concept in his article, "As We May Think." Five years later, Turing (1950) proposed a test for determining whether a machine could think like a human being. During the period of 1955-1960, growing with the progress of computer technology, AI came to its formative years. As mentioned earlier, Allen Newell, J. C. Shaw, and Herbert Simon wrote the "Logic Theorist," a program that simulates human thought . This was considered the first AI program in the United States. Another contribution during the decade was the symbolic computer language, LISP, devised by John McCarthy in 1958. LISP is one of the most frequently used languages in AI.

Since the 1960s, there has been a big transformation in AI. Bielawski and Lewand (1991) stated:

> Newell, Simon, and Shaw especially urged that researchers ask not *how* humans do what they do, but rather *what* they do. Scientists began to think that computers could be made to do what humans do even if the machines and the humans do not carry out the tasks in the same way. In short, the final result, and not the method, become the goal in creating machines that mimicked human behavior. And how do humans behave? By processing information symbolically, these researchers agreed. The new thrust in artificial intelligence, therefore, involved building hardware and developing software with the capability for symbolic manipulation. (p. 275)

From 1961 to 1970, AI researchers concentrated on human problem solving, heuristics, and robotics. In 1965, Lederburg and Feigenbaum developed a program named Dendritic Algorithm (DENDRAL) (Feigenbaum, Buchanan, & Lederberg, 1971). By analyzing mass spectrographic and nuclear magnetic resonance data, DENDRAL can infer the structure of an unknown chemical compound. DENDRAL was recognized as a landmark program in AI research. It was the first expert system which focused on specific human knowledge rather than general problem solving.

Other important outcomes of the 1960s were hypertext and hypermedia. Hypertext is based on Bush's idea that documents could be organized and accessed in non-sequential ways. As graphics and sound were added to this concept, the term hypertext evolved into hypermedia. The better known projects in the field were Engelbart's Augment and Nelson's Xanadu. Augment is an information repository system. It offered an early paradigm for hypermedia and used a mouse as its input device. Nelson coined the word "hypertext" and believed it could be used to organize a huge information base. Xanadu created a network environment in which people could interact with each other to create documents and audio and video media (Bielawski & Lewand, 1991).

Encouraged by the success of DENDRAL, many large expert systems were developed and marketed during the 1970s. The efforts on problem-solving theory and research were almost abandoned. Heuristics, or "rules of thumb," were incorporated with reasoning technique to construct practical systems. Three systems developed during this period were PROSPECTOR, MYCIN, and Expert Configurer (XCON). All three systems are still used today.

PROSPECTOR was a domain-independent consultation system developed in 1970 to assist geologists working on mineral exploration. In the system, the geological knowledge base and the mechanisms that employ this knowledge were separated (Duda, 1979).

MYCIN, begun in 1972 and completed in 1976, was one of the first expert systems to use probability-style reasoning for advising physicians on findings and diagnoses in the area of infectious diseases of the blood. The rule-based system included 500 rules in its

knowledge base. The project was initiated by Feigenbaum and developed by Shortliffe and his colleagues at Stanford University in the 1970s (Shortliffe, 1976).

XCON is a computer configuring expert system. The initial 800-rule system was completed in October, 1979 and extended to over 3000 rules by 1983. Before it reached its performance goal, XCON had offered a significant contribution to Digital Equipment Corporation by saving thousands of human-hour costs. XCON is one of the living commercial expert systems, and it has grown incrementally (McDermott, 1982).

Natural language processing was another innovation of AI in the 1970s. Why natural language study? There were two reasons. First, it was hoped that the study might result in the creation of real artificial intelligence; second, to enable machine-translation. Both had been dreams since the dawn of computer history (Chang, 1985).

Some projects, funded by Advanced Research Project Agency (ARPA), that focused on speech understanding systems made a great deal of progress between 1971 and 1976. Schank's Conceptual Dependency Theory provided one of the useful techniques in understanding natural language (Schank, 1972). Related research such as morphological/syntactic analysis and semantic/pragmatic analysis continued in the 1980s. Conceptual Dependency Analyzer, which applied Schank's theory, has been successful in language analysis (Chang, 1985). During the period of 1979-1981, Schank and his colleagues generated a series of natural language understanding systems: Fast Reading, Understanding, and Memory Program (FRUMP), Integrated Partial Parser (IPP), Better Organized Reasoning and Interface System (BORIS), and Computerized Yale Reasoning and Understanding System (CYRUS) at Yale University. These systems were able to summarize the stories that they understood in several languages (Schank, 1984). In the 1980s, the study of natural language processing increased in the United States and grew in other countries with different languages. Several experimental bilingual machine-translation systems have been reported. In the mid-1980s, a few unsophisticated domain-limited translation systems were marketed even in Japan (Chang, 1985).

Knowledge Engineering

The production system has created a practical experiment of a language recognition process and accelerated the establishment of knowledge engineering. What is knowledge engineering? According to Feigenbaum's definition, knowledge engineering is the process which "involves domain experts and computer scientists working together to design and construct the domain knowledge base" (Barr & Feigenbaum, 1981, Vol. II, p. 84). In other words, it is an applied science that aims at simulating part of human cognition through computers, constructing the knowledge structure, and working on the reproduction of intelligence.

In the last decade, knowledge engineering has been widely used mainly because studies in artificial intelligence such as Knowledge Representation Language (KRL), commonsense algorithm, frame structure, and production systems have greatly enriched the connotation of knowledge engineering. Knowledge-based systems or expert systems are the results of combining the above theories and developing them into applied technology.

Knowledge-based System

The most successful technique which applied knowledge representation schemes probably is the knowledge-based system. Knowledge-based systems sometimes can be considered development tools that can be used to build a set of programs called an expert system to solve problems that normally require the abilities of human experts; therefore, an expert system that captures the knowledge of domain experts can make the specific knowledge available to novices or less experienced users.

In general, a knowledge-based system requires at least three major components: (a) a knowledge base, (b) an inference engine, and (c) a developer interface. The knowledge base consists of goals, rules, and facts about a domain of expertise. The inference engine is responsible for reasoning and strategy control. The developer interface is a special design method or area by which the system can be connected or can communicate with the user, other systems, and devices (Chang, 1987b).

Although a knowledge-based system provides very limited knowledge, its problem solving is similar to human problem solving in some ways. For this reason, rule representation and production system structures have been used as the backbone of some

expert system such as DENDRAL, MYCIN, and PROSPECTOR (Barr & Feigenbaum, 1981). Expert systems composed of production rules are also called rule-based systems, and they are the most popular expert system structures.

Expert System

An expert system is a computer program that uses knowledge, facts, and reasoning techniques to solve problems and make decisions. Although PASCAL has been used to construct expert system inference engines (Reasor, 1985) and BASIC can be used to build MYCIN-like expert systems (Grigonis, 1987), choosing any high-level programming language as the main development tool for expert systems probably is not the best idea. Since Essential MYCIN (EMYCIN) successfully separated the inference engine from a knowledge base (Melle, Shortliffe, & Buchanan, 1984), several powerful rule-based expert system shells have been developed by software companies. These tools allow the knowledge engineer to create the knowledge base in an adaptive way and to group rules effectively. Many applications of expert systems have been developed by corporations and research institutes since 1984. Westinghouse applied an expert system to nuclear power plant design and General Motors developed a series of expert systems for mechanism design and management purposes (Fersko-Weiss, 1985).

Expert System Development Stages

Research on building expert systems or knowledge engineering methodology focuses not only on knowledge representation but also on extracting and organizing knowledge from domain experts. In a sense, knowledge acquisition is the principal barrier in the development of expert systems. A knowledge engineer is responsible for designing and building an expert system such as identifying problems, acquiring knowledge from human experts, coding the explanation of reasoning, determining the inference strategy, and developing a system that simulates the expert's problem solving. To fulfill these challenges, the knowledge engineer serves not only as a system constructor but also as a coordinator among human experts, computers, and system users.

To construct an expert system, there are five stages in knowledge acquisition: identification, conceptualization, formalization, implementation, and testing (Hayes,

Waterman, & Lenat, 1983). It is an easy concept to accept in principle but, sometimes, an abstract one in practice. As each system has specific purposes and limitations, these stages may vary from one individual situation to another. Meanwhile, although these stages can be identified in sequence, there are no clear distinctions between the stages.

Computer-assisted Instruction

Computer-assisted instruction (CAI) is a growing application of microcomputers in education. A wealth of literature exists on the subject. The term "computer-assisted instruction" normally covers a wide range of uses of computers for instructional and educational purposes. The Programmed Logic for Automation Teaching Operators (PLATO) project, which was begun in the early 1960s at the University of Illinois, can be considered a successful CAI system (Bitzer, 1986).

Numerous CAI computer software programs have been developed in the last decade. Earlier programs for the Apple and TRS-80 microcomputers were drill-and-practice or tutorial types of instruction tools for elementary and secondary education. School libraries and media centers have been quite active in using these programs to teach library research skills (Gratch, 1986). Simulation and modelling software were developed for use in teaching physics at the undergraduate level (Boardman et al., 1988). More recently, academic libraries have provided CAI programs to instruct freshmen students in the use of library resources (Lawson, 1988). A variety of language-learning programs have been used in the field of Computer-Assisted Language Learning (Jones & Fortescue, 1987). Many other applications are designed for commercial and industrial training purposes (Matta & Kern, 1989).

The increase in CAI application, coupled with advanced technology and full-market promise, provides a vision and challenges for creative efforts. Several computer languages such as BASIC, PASCAL and C are used in designing CAI software. Another method which is designed specially for producing educational software is authoring language. Authoring language provides certain features that allow branch functions. An outstanding example is PILOT, which many teachers use to develop their own courseware. Authoring systems are programs designed for the purpose of constructing teaching materials or tests in

the simple formats such as multiple-choice or true/false questions. As authoring systems do not require much knowledge of computers and programming technique, the teacher can generate instructional materials to be presented by a computer and the computer simply follows the predesigned instructions of the teacher in interacting with a student. Learning how to operate authoring systems is much easier than learning how to program authoring language (Dever & Pennington, 1989).

Instructional modes such as electronic books, drills, simulations, and games have been used in CAI software. Experiments showed that computer-based instruction can promote children's creative thinking and problem-solving skills (Papert, 1970). Much CAI courseware, however, is concerned with the strengths and limitations of the system itself, rather than with educational aspects (Elsom-Cook & O'Malley, 1990). For example, a few CAI creators simply use authoring language to transfer testing or teaching materials from some other media onto the computer. This courseware may be dogmatic and harmful.

Interactive Multimedia

Multimedia typically refers to the combination of computer graphics, animation, optical storage, and image and sound processing (Chang, 1991). With the advent of modern technologies, it is possible to integrate many types of media, text, graphics, video, and audio into one package. This package could be assembled from a very wide variety of information sources: hardcopy, slides, film, audio compact disks, video tapes or disks, and pictures from a video camera (Ciser, 1990).

> The principal investigator and project director of PROJECT EMPEROR-I said:
>
> Multimedia/hypermedia's forerunner is hypertext. The concept of hypertext has been with us since the 1940s, yet it has been brought down to the “household” level only in the last couple of years. Particularly since the introduction of Apple's Hypercard in late 1987, we seem to be entering a new chapter of hypertext / hypermedia information delivery. In a short twenty-month period, there has been a quantum surge of interest in hypermedia applications. Indeed, behind the complex and quite confused HyperWeb environment, instead of accessing, retrieving, delivering, and utilizing print-based information only, the technological environment

> is ready now for us to be very demanding and aggressive in seeking needed information which is available in all forms and formats. In other words, we want to access easily and quickly the massive amount of multimedia information *as we think*. We are in a hypertext/hypermedia age! (Chen, 1989, p. 2)

In June of 1986, Ambron and Hooper (1988) organized a conference on multimedia in education and emphasized that multimedia can improve the quality of education in two ways: (a) teachers will be able to demonstrate difficult concepts by having the ability to access information and the ability to illustrate ideas with the combination of visual, audio, and text, and (b) students will have a new way to communicate and learn from a wide variety of resources.

Some research applying multimedia technology in education has been reported. In 1989, the Department of Defense (DoD) conducted a study on the use of interactive videodisc technology in training and education as it pertains to effectiveness, cost-effectiveness, time on task, retention, and overall applicability to current and future DoD training and education requirements. In July 1990, the abstract of the final report stated:

> In response to Congressional direction, a quantitative, analytical review (a "meta-analysis") was completed of interactive videodisc instruction applied in Defense training and in the related setting of industrial training and higher education. Over all instructional settings and applications, interactive videodisc instruction was found to improve achievement by about 0.50 standard deviations over less interactive, more conventional approaches to instruction. This improvement is roughly equivalent to increasing the achievement of students at the 50th percentile to that of students currently at the 69th percentile. An improvement of 0.38 standard deviations was observed across 24 studies in military training (roughly an increase from 50th to 65th percentile achievement). An improvement of 0.69 was observed across 14 studies in higher education (roughly an increase from 50th to 75th percentile achievement). Interactive videodisc instruction was more effective the more the interactive features of the medium were used. It was equally effective for knowledge and performance outcomes. It was less costly than more conventional

> instruction. Overall, interactive videodisc instruction demonstrated sufficient utility in terms of effectiveness, cost, and acceptance to recommend that it now be routinely considered and used in Defense training and education. (Fletcher, 1990)

This report indicated that interactive videodisc instruction may be more effective and less costly than conventional instruction. Another study of multimedia is National Center for Supercomputing Applications' (NCSA) Video Macintosh, a video production system in NCSA Numerical Laboratory at the Beckman Institute. This desktop system was designed to be easy to use and easy to replicate. Users can apply NCSA Video Mac to create their own frame-accurate scientific visualization videotapes on the desktop and leave the lab, tapes in hand, ready for a meeting or presentation (Walsten, 1991).

Although interactive multimedia provides a way to combine computer graphics, animation, and audio effect into an interactive system, a few technical problems in multimedia technology still exist for personal computer users, such as adequate but unspectacular graphics and insufficient storage space for video images (Miller, 1989). In addition, the teacher still needs an effective, inexpensive, and programming-free tool to evaluate the student's performance.

Intelligent Tutoring System

Conventional computer programs solve problems numerically, follow a fixed algorithm, and mix control strategies with domain-specific knowledge. To run efficiently, the conventional program requires complete information as input data. It also requires a human to solve the problem before the computer does. In addition, its structure is difficult to modify. Unlike conventional computer programs, an intelligent system can solve problems symbolically and use general inference procedures rather than fixed algorithms. Besides, as control strategies are separated from domain-specific knowledge, it is flexible, and easy to modify (Lu, 1989). In other words, intelligent systems purport to emulate the human thinking process or, in a more accurate sense, to simulate human problem-solving ability.

An application program that provides insight into the current state of intelligent systems development is an intelligent CAI authoring system, Object-Based Intelligent

Editor-1: Knowledge-Based Editor (OBIE-1:KNOBE), developed by Freedman and Rosenking (1986). This system is a set of knowledge-based tools that enable authors to develop interactive simulation for computer-based training.

> OBIE uses a hierarchical frame-based scheme for this representation; the frame, consisting of slots denoting the device name, its states, and the values its states may have, together with appropriate text, graphics, and relative coordinates, is what we have been calling an "object." This representation is convenient for knowledge acquisition tools, since frames allow for 'default' slot denotations. (Freedman & Rosenking, 1986, p. 37)

The idea of using expert system technology in education is not completely new. A few intelligent systems have been used during the past ten years. GUIDON, from Stanford University, is an intelligent tutoring system in the medical domain. It trains students in diagnosis, and is built on MYCIN (Clancey, 1987). Another project aimed at developing a generic tutor, called Meno-tutor, hoped to achieve some degree of generality (being able to tutor in different domains) by vertically distinguishing between different discourse planning levels (Duchastel, 1989). At the University of New Hampshire, an intelligent tutoring system, based on a model of Intelligent Teaching Consultant (ITC), was designed as a collection of expert systems that can generate and debug programs and consult with the student about programs and debugging (Johnson, Bergeron, & Malcolm, 1990).

In the United Kingdom, the Salford University Physics Department engaged in a major project aimed at developing simulation and modeling software for use in teaching physics at the undergraduate level. Two of the most important aspects of the project were the user-interface for the programs and their distribution. A well-defined programming strategy has been developed for the project, based on the experience of over fifteen years' involvement in computational physics and computers in physics teaching (Boardman et al.).

In Belgium, research on developing a computation tool with which to teach reading skills in a foreign language was presented in 1990. The tool consists of three main elements: a program which merely displays reading material, a dynamic dictionary, and a simple augmented translation network (ATN) parser, which, together with the dynamic dictionary,

forms an expert reading system to be used as a trouble-shooting facility by the students (Nyns, 1990).

Summary of the Literature

The issue of whether language influences thinking or the other way around has caught the attention of scholars and experts from a variety of fields. As a result, the focus of study has shifted from the nature of human thinking to the relationship between language and intelligence. Artificial intelligence and knowledge engineering are pioneer attempts to bring theory into practice in this field.

Modeling human intelligence has been one of the purposes of artificial intelligence research since computers were invented. From the history of AI research, the main efforts can be divided into four major branches: natural language processing, computer vision and image interpretation, robotics, and expert systems. Of all the branches of AI, expert systems probably are the most sophisticated and practical technique.

In the conventional CAI system, the teacher uses an authoring language to develop courseware. Authoring systems allow people who have not had much programming experience to produce computer-assisted instruction with a limited set of functions. Some sophisticated authoring systems are powerful enough but somewhat more difficult for novices to use. Though authoring software permits the designs of screen display, answer-judging, record-keeping, and branching, most of this software lacks an interactive nature (Dever & Pennington). Besides, it is expensive and requires a lot of effort to learn how to use.

Applying expert system techniques to computer-assisted instruction has been in progress for several years. Like other expert system applications, the author of a tutorial expert system can change the subject domain of an instructional system without having to write a new one. This technique promises benefits such as being able to change the courseware without having to revise the program and modify instructional performance.

Limited research has indicated that multimedia may be an efficient tool for converting information into knowledge (Newhard, 1987) and that intelligent CAI would be the way to convey human knowledge to a student in an effective manner (Rambally, 1986). As Matta and Kern stated, "CAI can be viewed as potentially the ultimate expert system. The

computer is not only utilized as a facility in which to maintain a sophisticated knowledge base. Rather, the computer must also be prepared to convey that knowledge base to a student in an effective manner" (p. 77).

The next generation of tutoring systems will be more intelligent and incorporate new techniques in knowledge acquisition and representation. From the perspective of education, the computer is not only a calculation tool or a facility in which to maintain information, but also a research tool for educators, which provides a practical experimental environment. With the proliferation of microcomputers, CAI, interactive multimedia, and AI techniques in educational institutions, it is possible to develop multipurpose instructional systems through expert system development tools. The age of integrating CAI, expert systems, and multimedia technologies into a variety of applications in education is emerging.

CHAPTER III

METHODOLOGY

> *The method of grasping knowledge based on learning and researching; learning is a lifelong process of accumulation and inheritance while researching is the process of observation, cogitation, and creation.*
>
> *Tung-Ying Chang*

The methods applied in this study were devised from expert system development stages, but emphasized its conceptualization, formalization, and implementation. In practice, these stages were transformed into basic design concepts and design procedures.

Basic Design Concepts

Using computers as tutor, tool, and tutee was one of the major ideas in the study. To function as a tutor, the system should be practical, effective, flexible, and expandable. To function as a tool, the system should be accurate, precise, reliable, and friendly. To function as a tutee, the system should provide the best experimental environment for the researcher to understand and manipulate CAI, expert system, and multimedia technologies.

In the development process, the following ten discrete principles were applied: selection of an IBM PC compatible computer for software development and user environment, consideration for developing the system by using an expert system development shell, choice of a high-level programming language to write auxiliary programs, course-independent design, employment of rule-based representation, use of hierarchical frames for organizing basic structure of the system, top-down design, design of external software access interface to incorporate with application software, creation of tutorial modules to perform instruction strategies, and use of graphics to present Chinese characters.

Selection of an IBM PC Compatible Computer for Software Development and User Environment

Microcomputers are available to users and are relatively inexpensive for schools or

homes. To expert system developers, it is important to find a machine that does the job, provides the best performance, and is affordable. The PC, rather than the Macintosh, was selected because the PC environment offers flexibility and standard expansion boards (Heid, 1991). In Taiwan, Macintosh's prices are higher than IBM PC compatible computers. Many local manufacturers make quality, low-cost PCs, while Macintosh is supplied only by Apple dealers, and there is no education discount for schools and students.

Generally speaking, the Macintosh is the better computer for users, but "better computer" and "best seller" are two different things. In Taiwan, at the current time, PC's users number far more than Macintosh's. Another concern was the Chinese input system. Most Chinese input systems were developed for PCs. Although graphics were used to present Chinese characters in the system, courseware authors still need a Chinese input system for generating Chinese characters.

Consideration for Developing the System by Using an Expert System Development Shell

For most projects, the reason for using an expert system development shell could include speed and reduction of the painstaking tasks of knowledge construction. Many shells offer consultation and development models. It is not necessary for expert system developers to design knowledge representation structure and build inference engines. Even with an expert system shell, there will be some programming. Compared to building the whole expert system in LISP or PROLOG languages, such programming is simple and easy. Sophisticated shells allow the programmer to become an expert system developer and provide more opportunity for experienced domain experts to transfer their expertise in an efficient way.

Choice of a High-level Programming Language to Write Auxiliary Programs

Because no single language will work for every programmer, a few shells provide an external language interface (XLI) to communicate with programs written in other languages. With the XLI function, an external program can be compiled to access system data. Even with an expert system shell designed to expedite development and written with careful considerations by experienced programmers, it will still be necessary to write some auxiliary systems or external programs using a high-level language.

Course-independent Design

For a general purpose tutorial tool, the system should be conducted with course-independent design; courseware should be isolated from the tutorial system. Like structured data, courseware or testing material can be organized as data files and invoked by the system when it is needed so that courses such as English, mathematics, and physics can share the same system.

Employment of Rule-based Representation

When dealing with a human expert, it is important to choose a proper way to communicate with the expert. The most natural way to extract human expertise and heuristics is with "IF-THEN" rules. Rule-based systems are one of the most efficient knowledge representation methods in expert system technology; although there are many disadvantages such as lack of a context dependency mechanism, low inference efficiency in complicated systems, and the limitation of rule numbers. Rules, however, are comprehensible, easily modified, and can be controlled with other rules. For a tutorial expert system, rules can be used to express "what-to-do" and represent "how-to-do" knowledge. These features probably make the system more like a human tutor.

Use of Hierarchical Frames for Organizing Basic Structure of the System

Frames are useful data structures for representing knowledge. The frame-based representation is good at constructing inherent structure in rules and data. A frame which relies on the concept of inheritance can be divided into several concept-dependency frames or can be threaded by other frames to form hierarchical structures. This organizing or categorizing technique simplifies a complicated concept and connects related pieces of information in a meaningful way. Because of its flexibility in representing context and control mechanisms, the advantages of frame structure can sometimes minimize the disadvantages of rule-based representation.

Top-down Design

To design conventional programs and expert systems, generalize the idea first; then refine it step by step. The system should be designed and implemented from top to bottom;

in the frame structure that means to design the main frame or the root frame first, then the second level, and so on.

Design of External Software Access Interface to Incorporate with Application Software

When dealing with complex real-world problems, none of the single tools is totally satisfactory. Integration features offer a cost-effective way of manipulating existing data or transporting the data to other systems. From the viewpoint of system development, an external software access interface not only bridges the gap between expert systems and conventional programs but provides many convenient means to meet different requirements of individual users.

Creation of Tutorial Modules to Perform Instruction Strategies

An intelligent tutorial system may consist of an expert module, a tutorial module, and a student learning module. To simplify the system design, the expert module and the student learning modules can be set by a human expert who is responsible for generating teaching and testing materials, explaining each problem-solving decision in order to assist a student in understanding how to solve it, and predicting the student's level of understanding and learning style. With the help of a human expert, a tutorial module may consist of the strategies, rules, and processes that govern the system's interactivity with the student. While the system is working, the tutorial module performs like a human tutor.

Use of Graphics to Present Chinese Characters

Text is one of the most important communication tools for human beings. The task

of processing text in the computer can be divided into three steps: represented in a computer by numbers, entered with a keyboard, and displayed on the monitor.

The English language, which has a simple writing system, can be encoded by an eight-bit binary system such as American Standard Code for Information Interchange (ASCII), and can be typed at the keyboard and displayed as letters or symbols on the screen.

Chinese, which consists of more than ten thousand hieroglyphic characters, is the most ancient writing system . Chinese input is different and difficult. A Chinese input system which applies a 16-bit, or two-byte binary system, takes more memory space and processing time than English input systems. Besides, it takes at least 4 to 5 key strokes to input each Chinese character. These barriers have become problems in developing Chinese software; but, at the current time, it is the only way to generate Chinese characters in the computer through keyboarding. Another problem is that most Chinese students are not efficient keyboarders. An expert system also needs more memory space to run than a conventional program. Designing an expert system with Chinese-English input will be impractical before finding an efficient Chinese input system.

According to an old Chinese saying, "a graphic is worth more than a thousand words." Ideally, the graphic enhances instructional and application purposes. Based on these considerations, instead of hooking with a Chinese input system, the use of graphics to present Chinese characters for the purpose of instruction is probably feasible.

Design Procedure

After identifying the appropriate problems, assessing the significance of the system, and forming the concepts of design, the formalization and implementation stages brought the system designer one step closer to programming, while the testing stage involved evaluating the system to improve it. The procedure discussed below is a concrete description of the three stages.

Resources Assessment and Tools Selection

Human resources

At least three persons are needed to build the system: the human tutor or expert, the knowledge engineer, and the system user. The expert offers teaching experience and assists

in the study. The human tutor provides teaching and material to test of the system. The knowledge engineer is responsible for system design, using the shell to transfer the necessary knowledge into computer readable form, and other programming. The user or system tester need not be computer literate to run the system and make suggestions.

Shell selection

Some textbooks offered criteria for shell selection such as appropriateness of the tool to the problems, effectiveness of the developer interface and user interface, integration capability with other programs, and delivery systems (Bielawski & Lemand, 1988). Generally speaking, principles and criteria are easy to understand. After identifying basic design criteria, the developer should know what kinds of tools are needed. The problems are: Where is the shell? How to get it? Is it affordable? How to evaluate the tool before you choose it?

The best way to evaluate a shell is to use it. Unfortunately, few software companies allow users to return products for refunds. In addition, software evaluation is time-consuming work. Software catalogs, commercial advertisements, and articles in journals were helpful in obtaining and selecting shells.

At the beginning of the study, the researcher relied on software review articles in *IEEE Expert, AI Expert, Personal Computing, and PC Week* for shell evaluation. Based on the considerations of cost, functions, documentation, user friendliness, and publisher's reputation, Texas Instrument's Personal Consultant Plus (PC Plus) was chosen as the system development shell.

PC Plus is an integrated rule-based expert system shell based on the EMYCIN program. In PC Plus, a knowledge base consists of one or multiple frames which include two major forms: parameters and rules. Parameters were used as the basic components. In a frame, a parameter is a structure that identifies or contains a bit of information needed to arrive at a conclusion. Rules define the relationships among parameters and determine how to use the information during a consultation. Goals list one or more special parameters determined by the inference during a frame instantiation in a backward-chaining knowledge base. The frame structure and properties are shown in Figure 3-1.

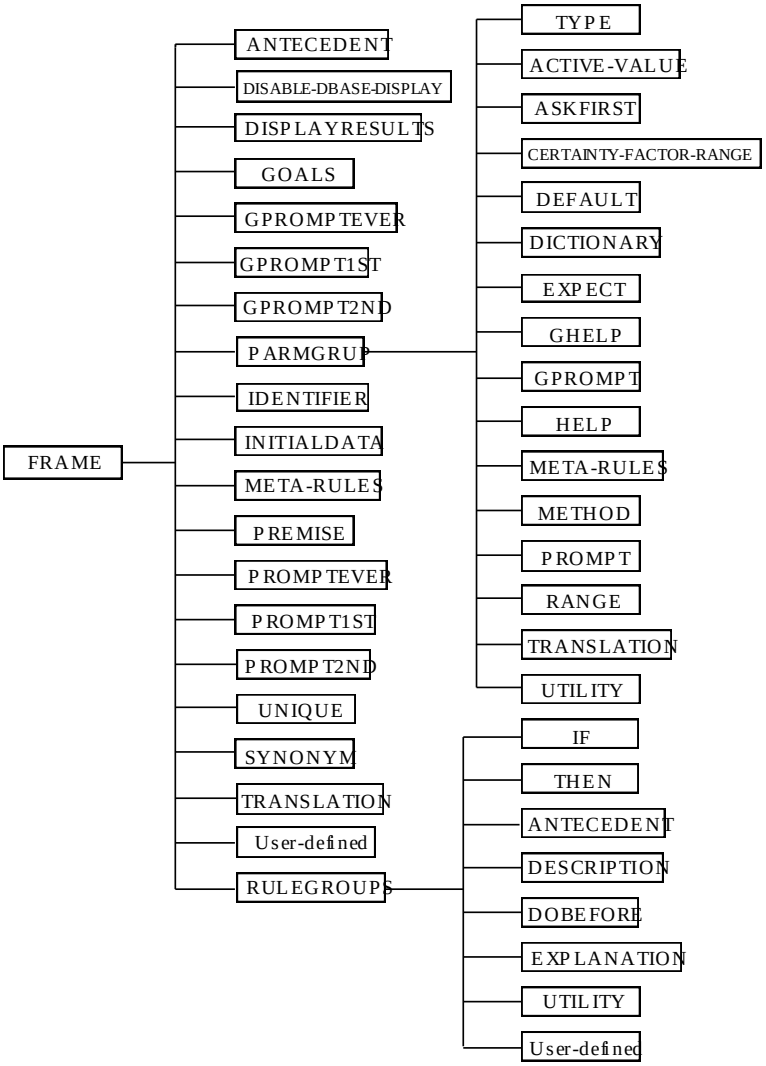


Figure 3-1. Frame structure and properties of Personal Consultant Plus.

The PC Plus inference engine is responsible for decision making: what data are appropriate to seek, the order in which to seek the data, and what rules are appropriate to use. Inference engines use received or derived information to decide how to process the next actions such as what parameter values to seek and what rules to try. Finally, the inference engine attempts to derive values for the goals. The primary control mechanism within PC Plus's inference engine includes backward-chaining and forward-chaining. The features analysis of PC Plus are shown in Figure 3-2.

Auxiliary programming language selection

LISP and BASIC have been used to develop expert system applications. PC Plus allows developers to use Scheme language to define functions for knowledge representation. The Scheme programming language is a dialect of LISP developed at Massachusetts Institute of Technology (MIT). Scheme has proved to be effective in developing expert system shells for microcomputers. BASIC language is appropriate for handling large amounts of text. In addition to the ability of string processing, BASIC has
many graphics statements and functions to create a wide variety of shapes, colors, and patterns on the screen. With graphics functions, the programmer can create vivid images to enhance teaching materials. Both GWBASIC and Quick BASIC can work very well with Chinese input systems. The high compatibility enables BASIC programs to generate Chinese hieroglyphs.

Software needed

ET Chinese System V 1.6, ETen Information System

Personal Consultant Plus V 3.0, Texas Instruments

Quick BASIC, Microsoft

Dr. Halo III , Media Cybernetics

Chinese Graphic Transfer System

Word processor & database programs for editing courseware and test material.

Features	Description
Knowledge Representation Schemes	Frame, IF-THEN rules, meta-rules

Other Knowledge Representation	Scheme, Scoop (object-oriented programming system)
Primary Inference Mechanism	Forward and backward chaining
Principal Knowledge Structures	Frame, parameters, and rules
Uncertainty Handing	Facts, rules
Logic and Mathematics	Boolean operating, float point, LISP expression
Facilities & Flow Control	Meta-rules, mapping functions, access method
User Response Format	Pop-up menu, on-line help, explanations, icon
External Language Interface	LISP, C, PASCAL, Assembly
External Software Access Interface	dBASE II, III, III Plus, Lotus 123, ASCII file
Rule Entry Language	Abbreviated Rule Language (ARL)
Developer Tools	Rule editor, logic tracing record, debug aid
Report function	Print, screen, file
Knowledge Chunk Limit	2000 rules (depends on RAM, size and complexity of the knowledge base)
Replay Capability	Review, Playback, New Start
Delivery Vehicle	Run-Time
Portability	C language
Graphics Program Interface	Use third party package, frame capture (SNAPSHOT)
Graphic Functions	Represent knowledges, conduct consultations
Graphics Support	CGA, EGA
Advanced Features	Garbage collection, fast-load files, autoloaded files
Advanced Add-on Package	Online, Image
Memory Support	Conventional memory, expanded memory, and extended memory
User Interface	Window-oriented

Figure 3-2. Features analysis of Personal Consultant Plus.

Hardware needed

The key to finding suitable hardware lies in looking for suitable software. After selecting the necessary software, the following equipment was needed:

IBM PC compatible computer with CPU Intel 80386SD-33MHz, 64 KB cached, 4MB on-board extension memory, 101 keyboard, mouse, 1.2MB-5.25" FD, 1.44 MB-3.5" FD, 80 MB HD, VGA 640x480 display card, and 14" color monitor, a scanner, and a HP LaserJet II Printer.

Pilot Test

After making sure that worthwhile problems had been identified, that the development environment was adequate, and that an expert system shell and auxiliary language had been chosen, the next step was to generate a pilot system. The advantages of generating a pilot system are that it:

1. Provides the opportunity to test the suitability of the development shell that has been selected.

2. Allows experiments to verify the important idea of design strategies, such as knowledge presentation scheme, frame structure, inference mechanism and instantiation control, and problem solving methods.

3. Gives the system designer a concrete concept of how to build and test an expert system. In addition, the designer will gain the confidence needed to handle a bigger system through the success of the pilot system.

Two pilot systems that were built before were too complex. The first pilot system was a bilingual children's disease diagnosis system based on the knowledge base which was revised from DOCTOR written by Edward Reasor (1985). In the pilot system, the knowledge base was presented in a graphic format. The purpose of designing this pilot system was to test image control and presentation ability. A Chinese hieroglyph data transfer program was designed to convert Chinese characters from black & white Hercules graphic to EGA color graphic. Figure 3-3 shows a graphic input function of the first pilot system. The user can input the patient's last hour temperature by pressing arrow keys to adjust the thermometric indicator.

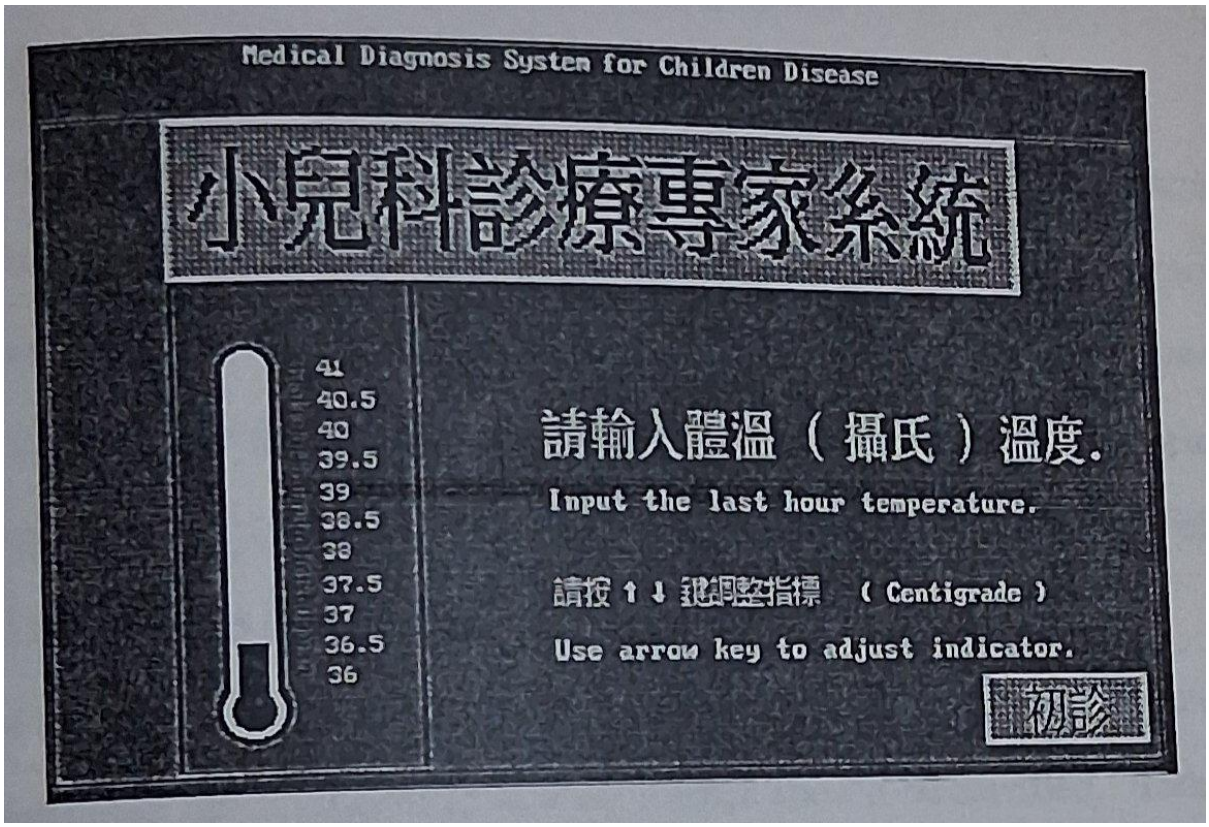


Figure 3-3. Graphic input function of the first pilot system.

The second pilot system was a welding procedure consultant system based on the author's experience of over five years welding technique in the Maanshan Nuclear Power Project. One root frame and four subframes were used in the rule-based system to include the necessary expertise and information. Forward chaining was used to reason the conclusion because the data and knowledge were extracted from mass technique information by a human expert. Besides, it was convenient to gather data and there were relatively few hypotheses to explore in the pilot system. Both pilot systems provided some experience to overcome programming barriers in the future and exposed some limitation of the selected expert system development shell.

Sketch Operating Flow Diagram

Flow diagrams are important and useful instruments for showing the sequence of operating processes and the relationships among modules. Analysis of the thought and the heuristics of the expert clearly helped to create rules and set up an inference mechanism. A refined, articulate flow diagram is absolutely necessary for system programming.

Building Knowledge Base

There are several steps in building knowledge bases. These are:

1. Model the input and output data structures. Like traditional programming, defining inputs and outputs of the system was an important step. Data construct occurs when two or more data elements are put together to form a larger data component. A record (in a traditional program) or a parameter (in a knowledge base) is meaningful data that can present a fact. A carefully organized data structure ensures accuracy of data processing and high program executing efficiency.

2. Set up the goal for each frame and identify the relation among frames. As each frame is independent, the relationship among frames should be clarified. Both vertical and horizontal relationships must be identified before setting up the goal of the frame.

3. Construct system architecture. A well-designed expert system is modular and expandable. In the system, the frame is the basic structure of a knowledge base. Knowledge is a library of information about an area of expertise. It is possible for separate knowledge bases to be linked to one another. In a large or complex knowledge base, a child frame can inherit the data from its parent frame, and any two individual frames can share the same data group. Careful analysis of the relationship among frames and placement of the frame in the right position to form the system structure is critical in the construction of a successful expert system.

4. Allocate executive operations. In the rule-based system, all reasoning is executed by means of frame instantiation. When knowledge is created, it is a static, abstract representation of knowledge and structure. After the system instantiates the frame, the frame is given a dynamic, concrete reality. To allocate the executive operation, mechanisms to cause and control instantiation must be built into knowledge bases.

5. Frame construction. In the PC Plus, each knowledge base must be based on a root frame. Additional frames can be added to the root frame. A frame is a collection of data and information. The information includes parameters, rules, variables, and other essential components. These components define the structure and operations of the knowledge.

Both parameters and rules are important components. Parameters contain information that the system uses to infer conclusions. Each parameter has a name, a set of possible values, and several properties. Rules are IF-THEN statements that express the relationships among parameters. Some of these properties can define or modify search strategies. Once knowledge bases were constructed, the programming work was done.

Editing Tool Design

One of the purposes of the study was to create a programming-free administration and instructional environment to enable the teacher to generate low-cost courseware for students. Teaching and testing material, however, must be generated in a computer-readable form. Therefore, a simple, easy-to-learn, easy-to-use, and flexible editing tool for the system was necessary for the courseware designer.

The editing tool is a subtle data management program with a full screen text editor written in Microsoft Quick BASIC. This program can be loaded on the system or operated independently.

Courseware Design

Before beginning the preparation of teaching and testing material in the system, several subjects were considered. English classes in Taiwan are popular and most students spend a lot of time and money to learn the language. An English grammar tutorial program designed to teach junior high school students about verb tenses was selected as courseware for system testing.

The courseware author, who had taught junior high school English grammar for the past ten years in Taiwan, designed a computer-assisted instruction program including instructional materials, multiple choice questions, and a courseware analysis document in Chinese. The courseware analysis document which explained the expert's knowledge, thoughts, and heuristics served as a communication tool between the expert and the knowledge engineer. This document was refined and represented in rule structure. Testing material such as idiom training, conversation practice, expression structure, sentence components rearrangement, and reading comprehension were also included. Each set of testing materials was identified by an unique serial number. All testing materials were

divided into two formats: the ASCII text file and the graphic file. ASCII text files were revised with editing tools or other word processors; graphic files were generated with the Dr. Halo package. Both the ASCII text file and the graphic file were given a DOS filename which was pre-assigned during the system development. As these materials were designed for the purpose of system testing, none of the curriculum instruction and testing material used in the study has been evaluated.

System Testing

Once the necessary courseware was completed, the system was ready for function testing or consulting. Testing covered each of its capabilities and the individual modules to see if the developed system achieved its intended goals.

Unlike commercial expert systems which are much concerned with market responses, the tests of the research focused on technological performance. From this perspective, accuracy, reliability, effective reasoning, user-friendliness, and run-time efficiency were used to test the system. Test items included input/output, text, graphics and animation effect, score and record management, external program access interface, screen design, and expected functions of the courseware editing tool.

CHAPTER IV

SYSTEM ANATOMY

We can see only a short distance ahead,
but we can see plenty there that needs to be done.

Alan Turing

System Overview

The developed expert system in the study is named Taiwanese Intelligent Tutorial Expert System (TITES). It offers a tool to enable teachers to generate teaching and testing material, to score tests, to record tutorial data, and to give instructions to students. TITES currently runs on IBM PC compatible computers and was packaged on four floppy discs: knowledge base disc, utility disc, courseware disc, and PC Plus runtime disc. As there are more than two hundred files in the package, it is recommended that the system be installed on a hard drive.

Operating Procedure

After entering "TITES" at the DOS prompt, the screen message shows that the system is loading the knowledge base, then the TITES system title screen appears as shown in Figures 4-1 and 4-2. To continue, press RETURN/ENTER key, and a user information form including a password request (see Figure 4-3) appears. The ID, age, and password are invisible when the user enters these data. Users must type each item and input an effective password before using the system. Lack of any item or incorrect password will result in a logon failure and invoke the system restart process.

When the user's password is accepted, the system will ask the user if introductions are needed; if so, a brief explanation of the system will appear on the screen. If the user skips the introductions, the system will show the functions available to the user. By using the arrow keys to place the color cursor over the function and depressing the RETURN/ENTER key, the user can choose instruction, examination, text file editor, or quit functions. The instruction function includes tutor, digest, simulation, drill, and animation; the examination function consists of five tests. Figure 4-4 shows the functional organization of TITES.

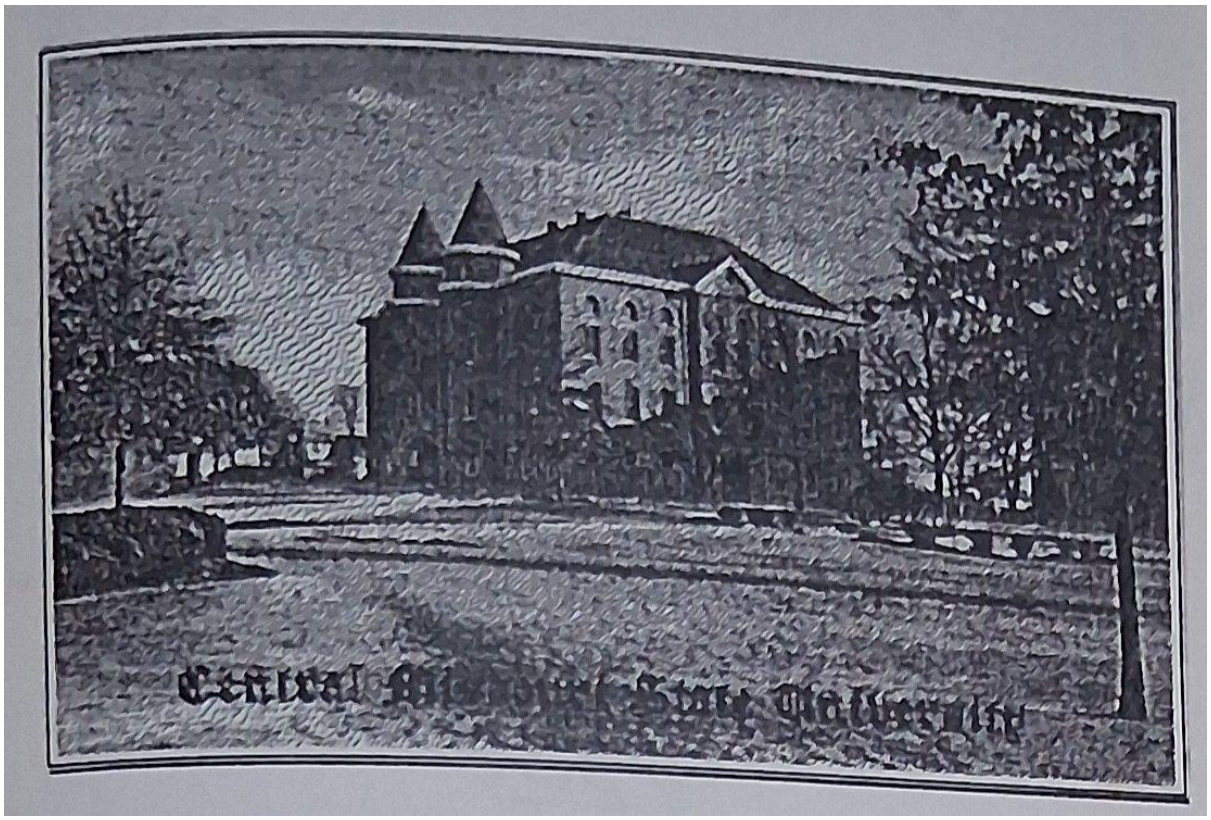


Figure 4-1. First title page of TITES.

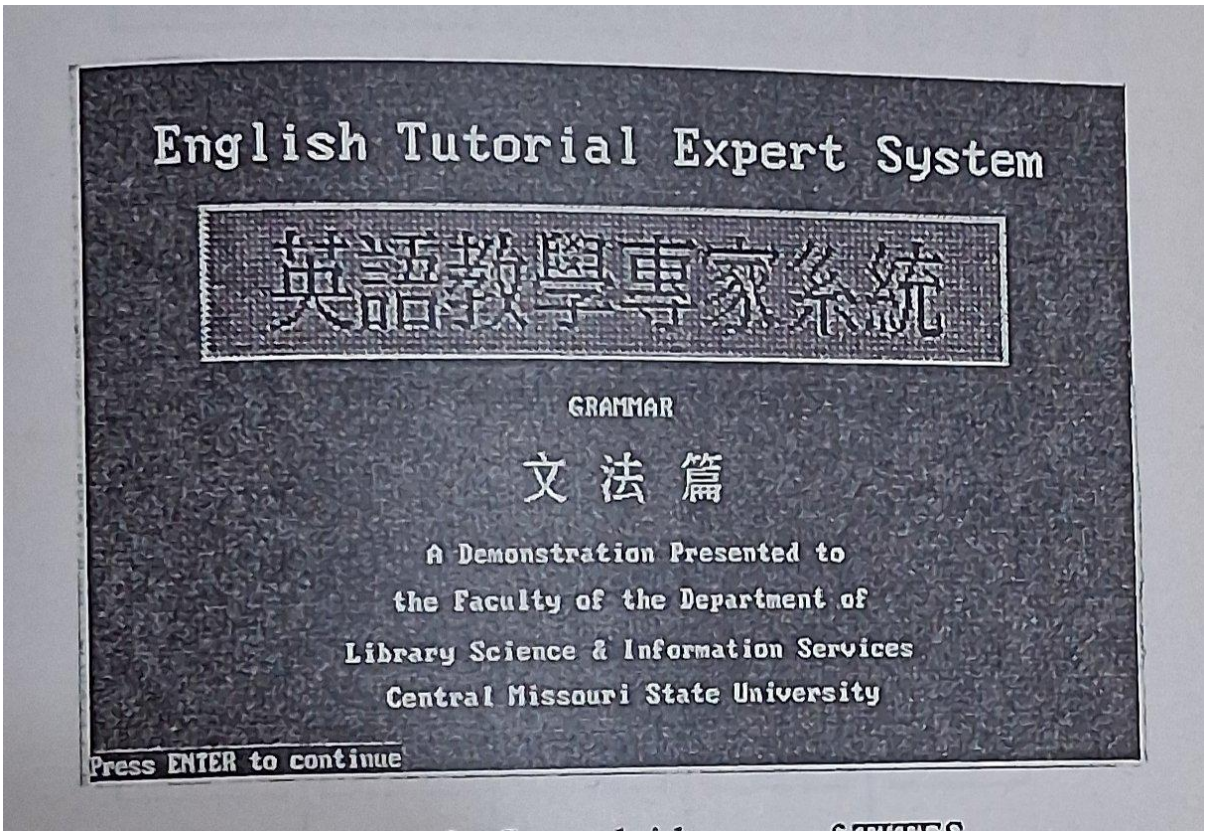


Figure 4-2. Second title page of TITES.

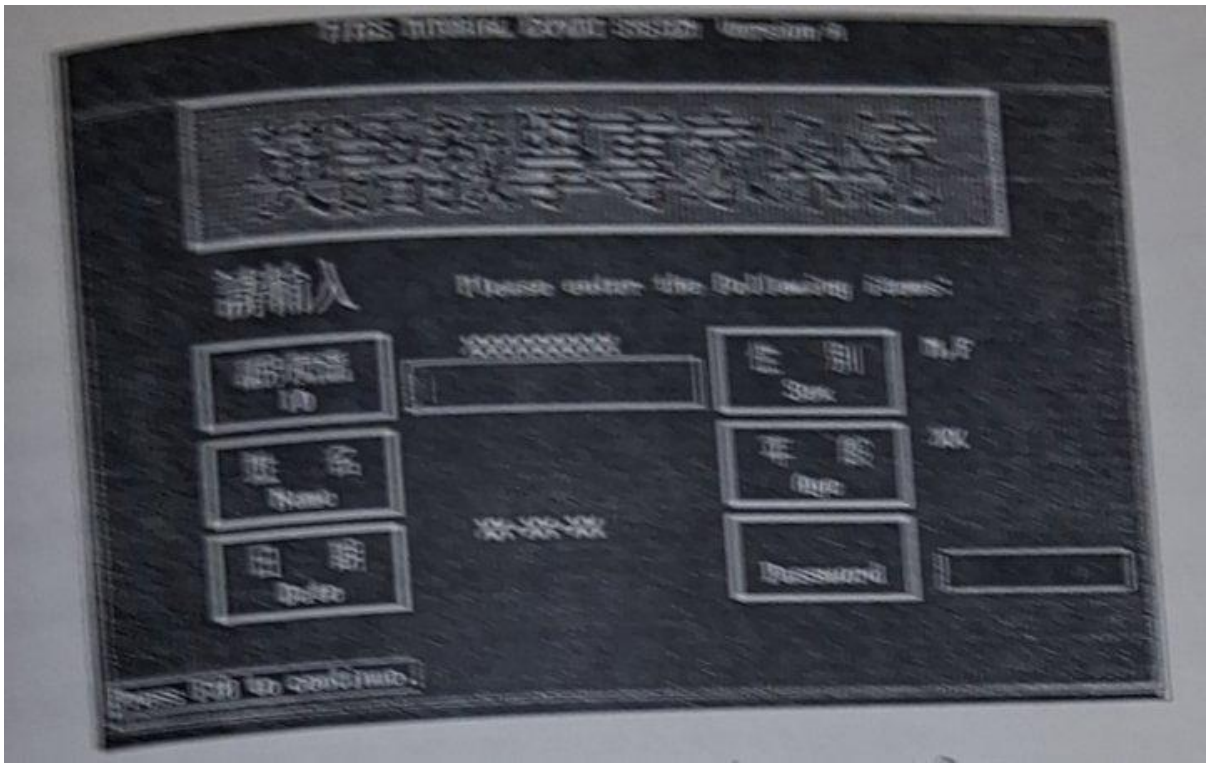


Figure 4-3. User information request form.

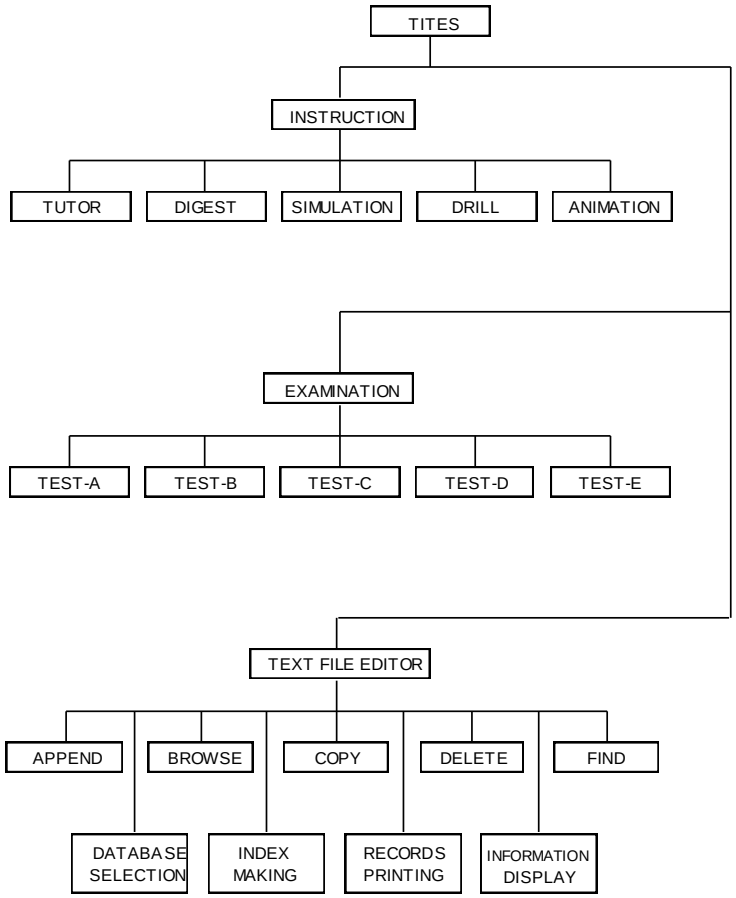


Figure 4-4. Functional organization diagram of TITES.

The tutor instruction is a book emulator which presents its contents on the screen as shown in Figure 4-5. When the contents page is displayed, the user can move the cursor and select the appropriate chapter. After pressing the RETURN/ENTER key, the title page of the chapter appears and shows the user how to use Home, End, and arrow keys to turn pages forward or backward. The user can also press the Esc key to return to the contents page for accessing other chapters at any time or to quit the current chapter. By selecting the tutor instruction, the user can read all the predesigned tutorial materials for that subject. The digest instruction is similar to the tutor instruction; however, the materials were condensed so that the user can browse them by flicking pages.

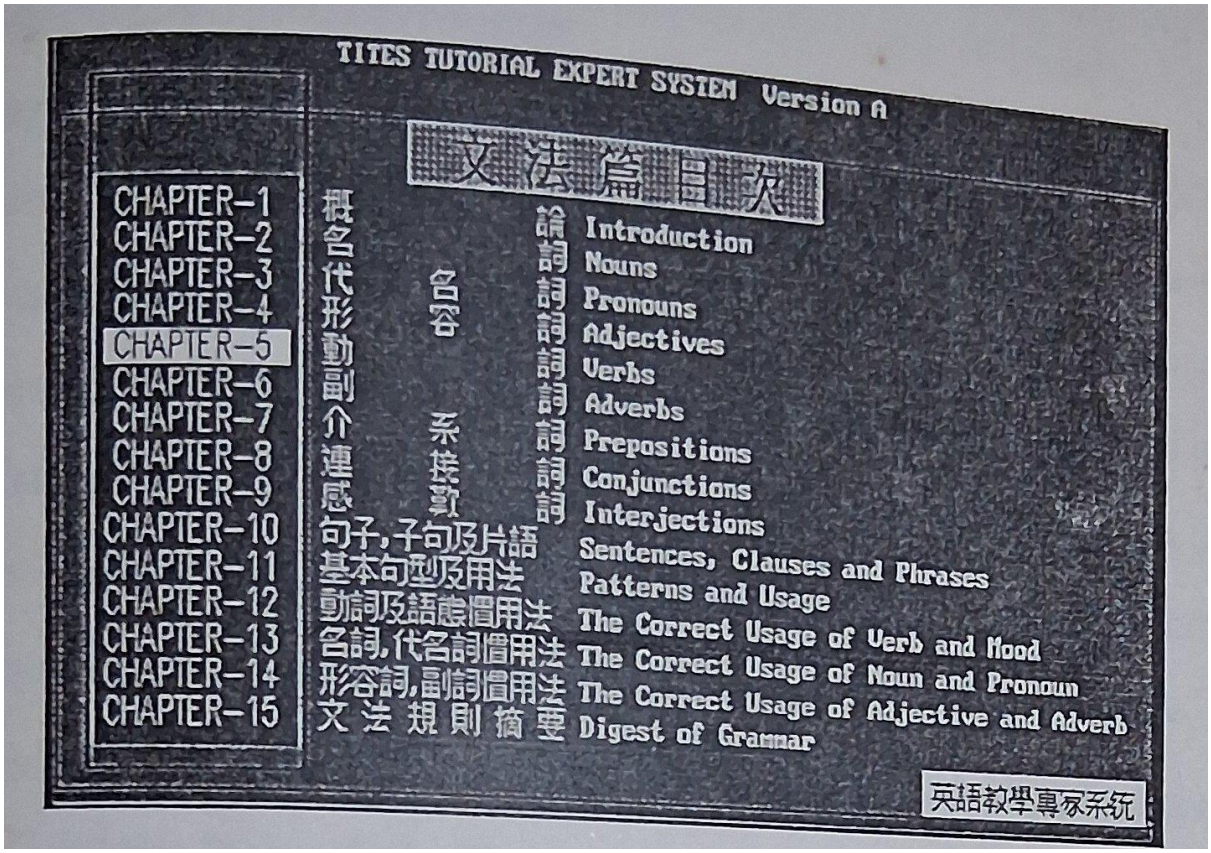


Figure 4-5. Contents screen in the tutor module.

The drill function is a hierarchical knowledge presentation which enables users to review an article or paragraph. The knowledge was compiled in a ladder type arrangement by levels of difficulty. If the user needs detail or advanced information, s/he can choose

terms or keywords from the screen. The first time a user invokes the drill, a series of topics appears; then the user can use keys to navigate horizontally or vertically.

The simulation instruction can be used in technique training. For example, users can strengthen writing skills through the essay writing simulation. When users choose a topic from the screen, several groups of sentences give step-by-step instructions to construct a paragraph. The user can select one of the sentences in each group or write an original sentence. After the user has constructed a version of the topic, s/he can read the paragraph on the screen and review the best version. The user can also invoke an instructional module about how to construct a paragraph from these sentences.

Animation is a set of motion pictures which can be used to present a scene, a fact, a phenomenon, or an idea. Users can choose options to improve their knowledge, gain better understanding, and promote retention.

The examination function provides five types of tests. Each set of tests includes ten questions which allows a user to select the right one from four possible answers. Test-A consists of text-based multiple-choice questions. When a user moves the cursor to TEST-A and presses the RETURN/ENTER key, a series of test numbers appears on the screen. The user can choose one of the tests or QUIT to return to the last screen to invoke other test types. After the user has chosen one of the tests and pressed the RETURN/ENTER key, a yellow color testing guide appears to explain how to give answers and prompts the user, "IF ready, press RETURN/ENTER key." Questions and possible answers appear on the screen one by one (see Figure 4-6). The user can type A, B, C, or D or use the arrow keys to move the cursor to select the right answer. After the user answers the ten questions, a score report including serial number, test number, score, quantity of right and wrong answer, and time used appears on the screen as shown in Figure 4-7. The user can review the questions and answers, and correct answers on the next screen through the playback feature (see Figure 4-8). Then the user can ask for tutorial instruction or select more tests to continue (see Figures 4-9 and 4-10).

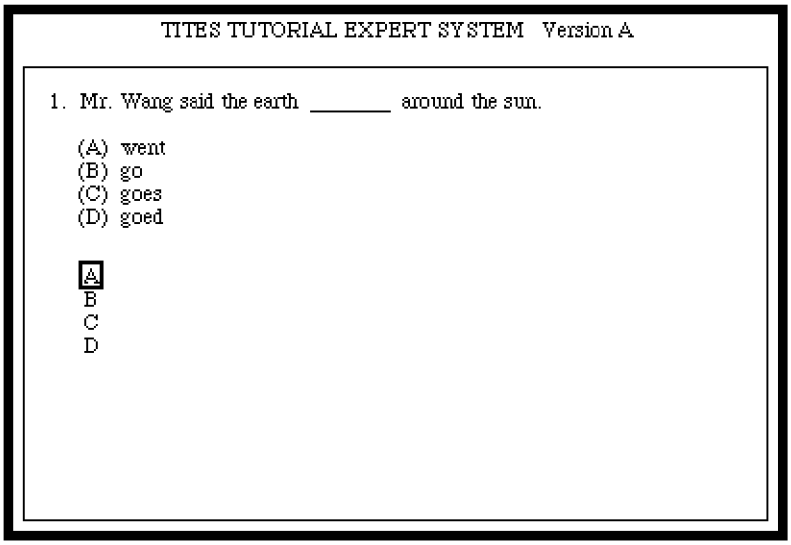


Figure 4-6. Multiple choice question of Test-A.

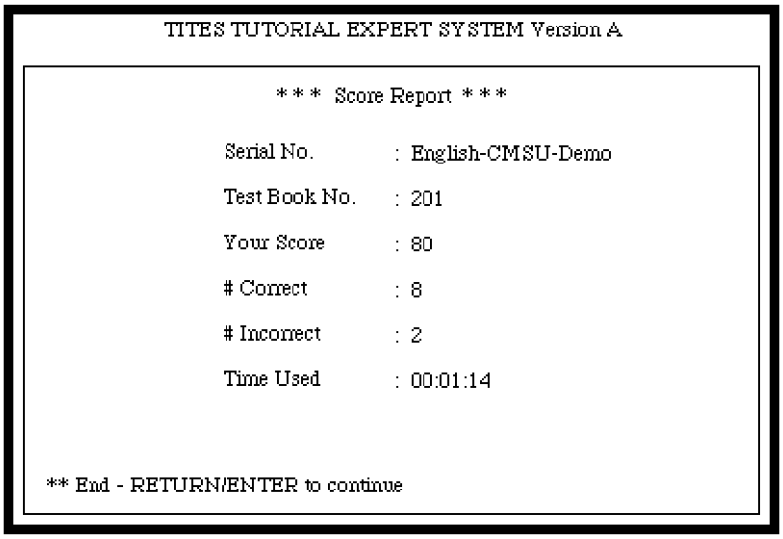

TITES TUTORIAL EXPERT SYSTEM Version A
*** Score Report ***
Serial No. : English-CMSU-Demo
Test Book No. : 201
Your Score : 80
Correct : 8
Incorrect : 2
Time Used : 00:01:14
** End - RETURN/ENTER to continue

Figure 4-7. Score report of the test result.

Test-B is a graphic-based multiple-choice test (see Figure 4-11). Test-B and Test-C are designed for Chinese language, mathematics, geography, physics, and other scientific questions that need graphics for demonstration. Each screen display shows one question. Test-C is also a graphic-based test. Each test offers ten blanks for users to complete (see Figures 4-12 and 4-13). Students can use the RETURN/ENTER key to choose the appropriate position to type the answers from the keyboard. All sentences used in Figures 4-12 and 4-13 were adapted from Robert James Dixson's *Essential Idioms in English* (Regents, 1971). Each sentence was coupled with relative Chinese translation.

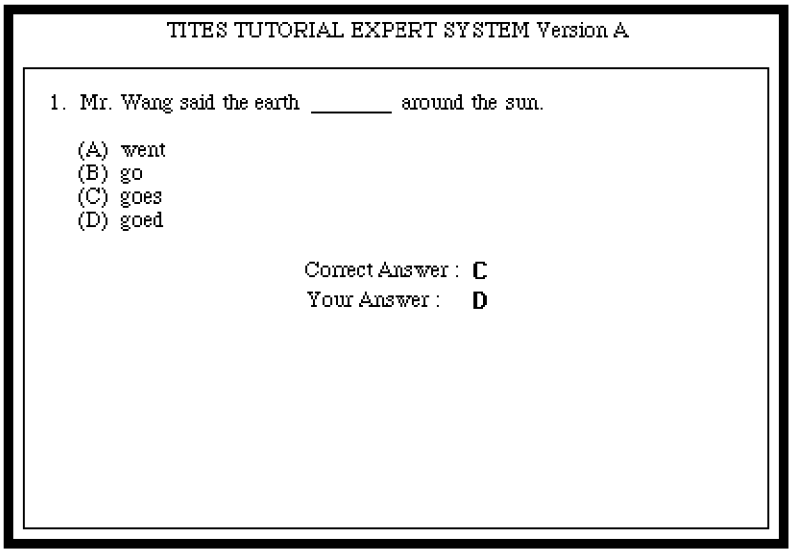

TITES TUTORIAL EXPERT SYSTEM Version A
1. Mr. Wang said the earth _____ around the sun.
(A) went
(B) go
(C) goes
(D) goed
Correct Answer : C
Your Answer : D

Figure 4-8. Screen showing a question, user's answer, and correct answer.

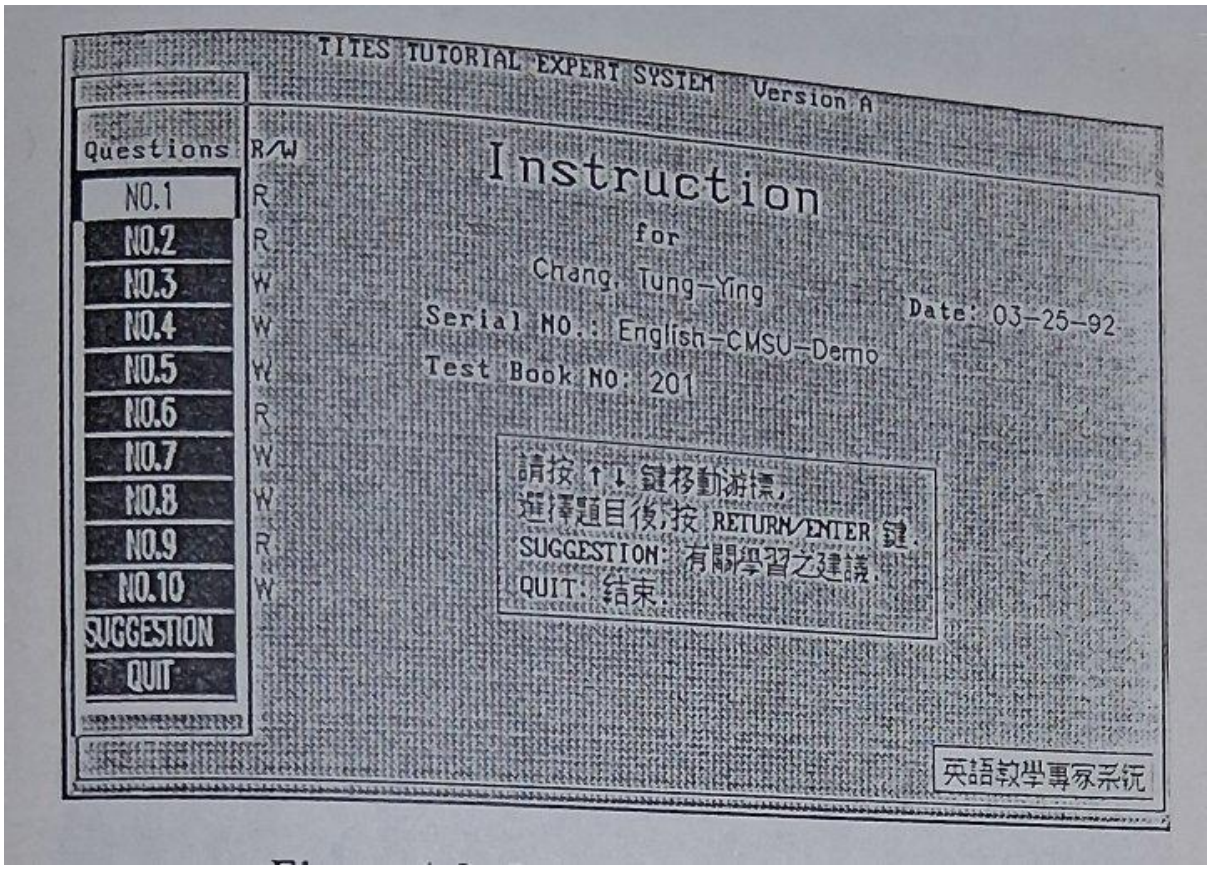


Figure 4-9. Instruction menu of Test-A.

Test-D is a test of sequential arrangements which can be used to test logical thinking, quantity analysis, and assembly techniques. Test-E is a reading comprehension test. Each test include two topics. Users read an article on the screen and answer the

questions about the meaning of the material. The reading material may include a wide variety of topics. Before learners answer the questions, they can press the F1 key and review the reading material from a window on the screen as shown in Figure 4-14.

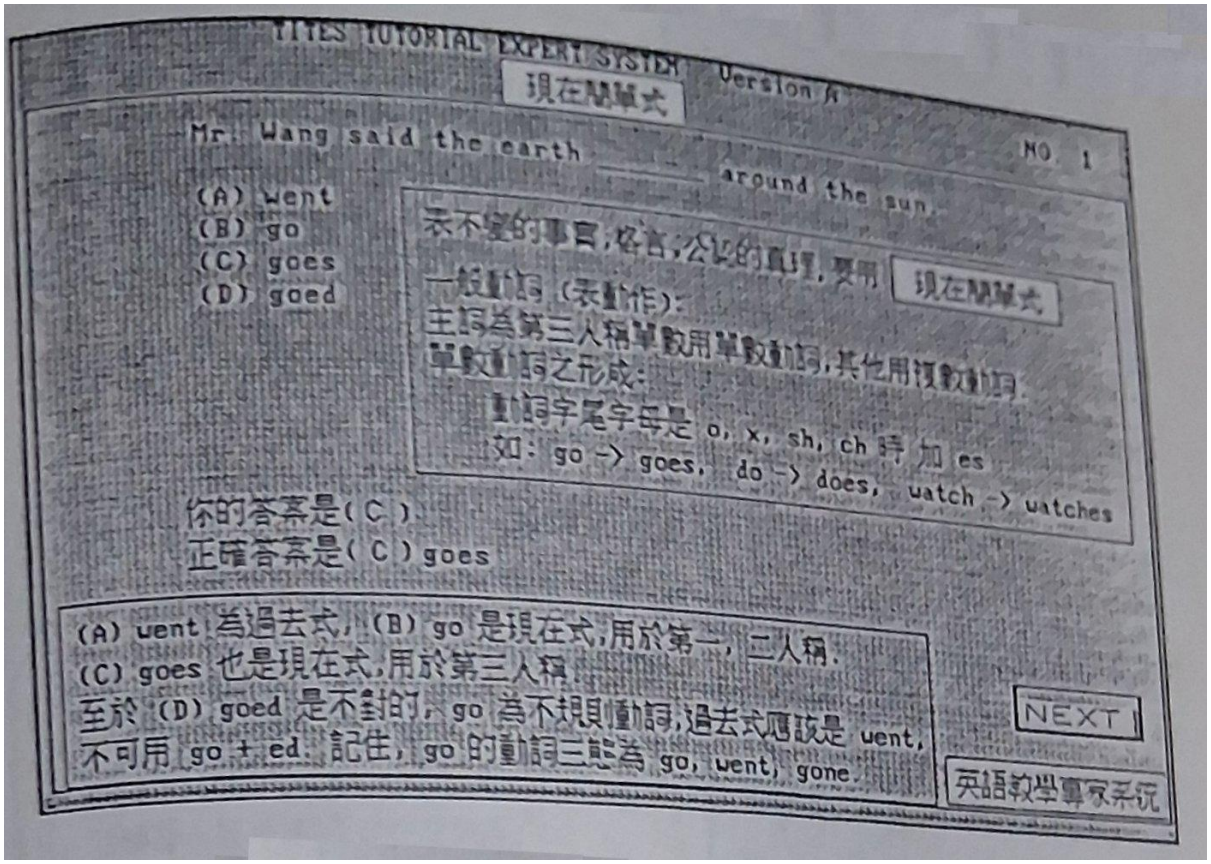


Figure 4-10. Instruction screen of Test-A.

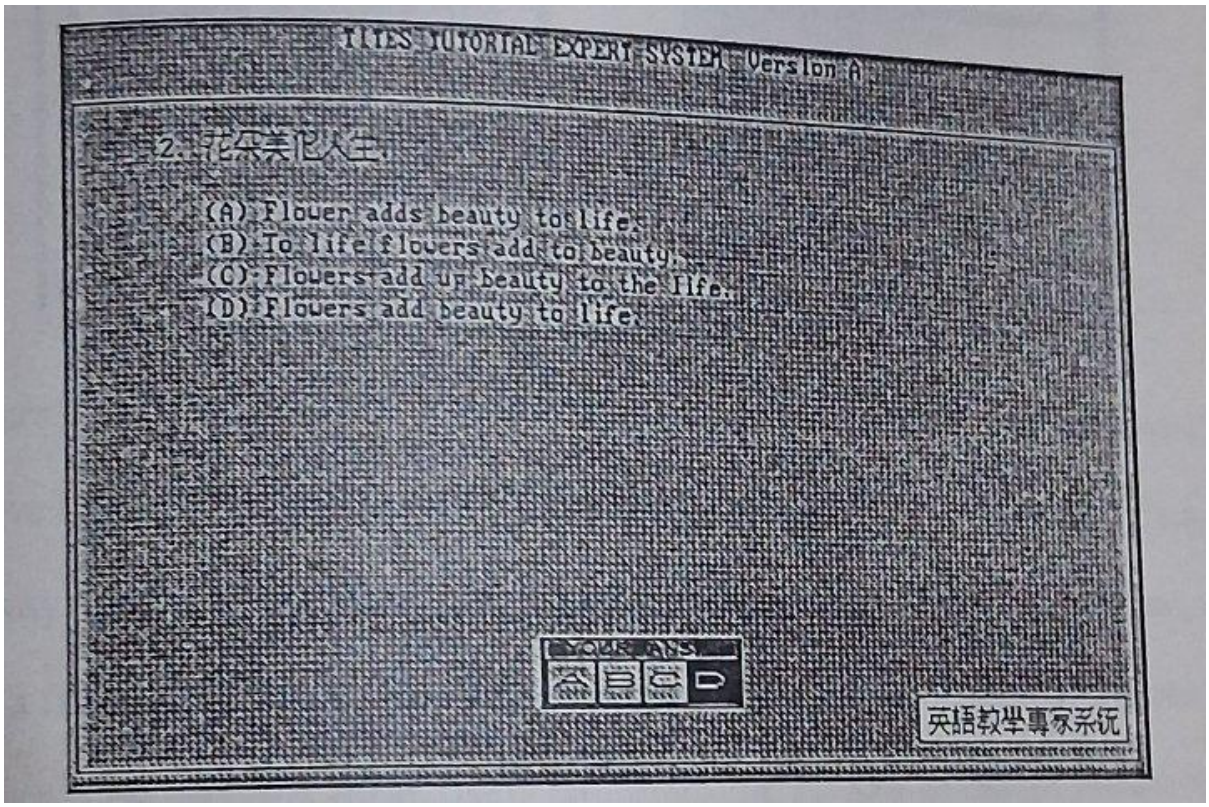

TITES TUTORIAL EXPERT SYSTEM Version A
2. 花点美化人生.
(A) Flower adds beauty to life.
(B) To life flowers add to beauty.
(C) Flowers add up beauty to the life.
(D) Flowers add beauty to life.
YOUR ANS
英語教學專家系統

Figure 4-11. Graphic-based multiple choice questions of Test-B.

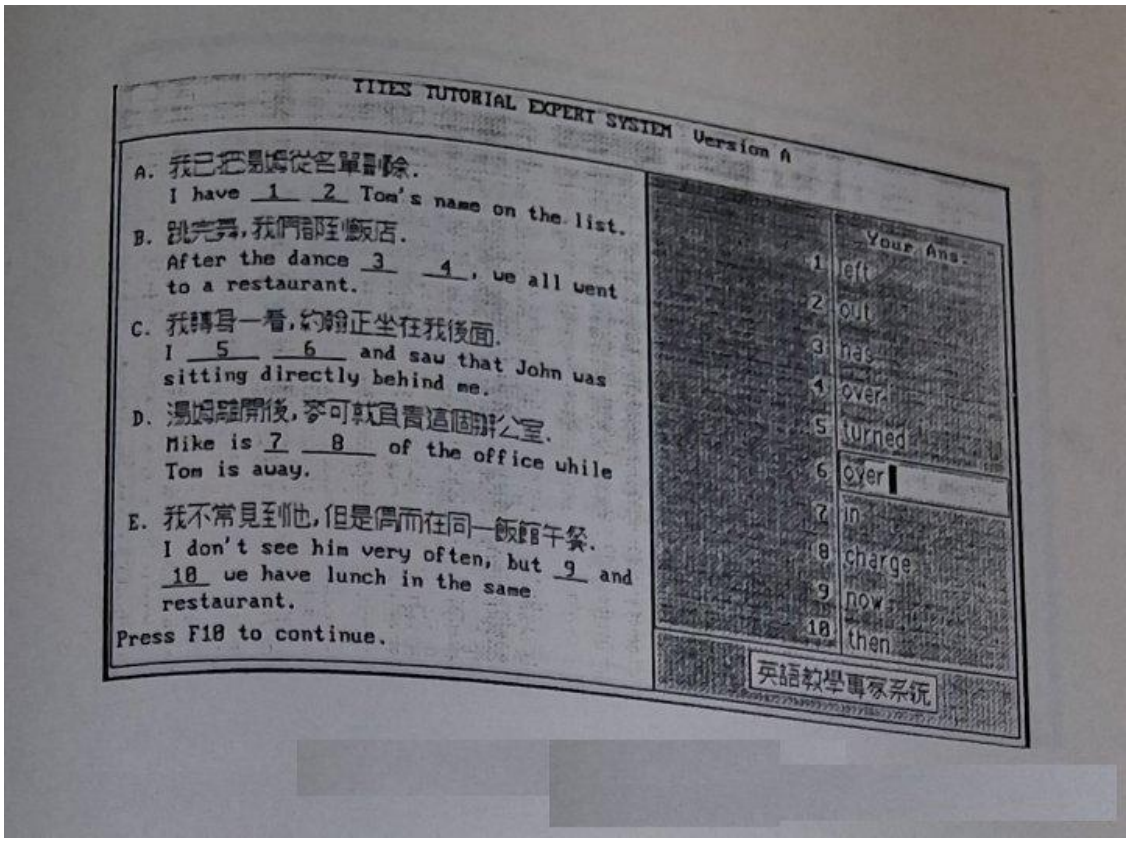

TITES TUTORIAL EXPERT SYSTEM Version A
A. 我已把湯姆從名單刪除.
I have 1 2 Tom's name on the list.
B. 跳完舞,我們都旺飯店.
After the dance 3 4, we all went to a restaurant.
C. 我轉身一看,約翰正坐在我後面.
I 5 6 and sau that John was sitting directly behind me.
D. 湯姆離開後,李可就負責這個辦公室.
Mike is 7 8 of the office while Tom is away.
E. 我不常見到他,但是偶而在同一飯館午餐.
I don't see him very often, but 9 and 10 we have lunch in the same restaurant.
Press F10 to continue.
Your Ans
1 left
2 out
3 has
4 over
5 turned
6 over
7 in
8 charge
9 now
10 then
英語教學專家系統

Figure 4-12. Test-C: Ten blanks for users to complete.

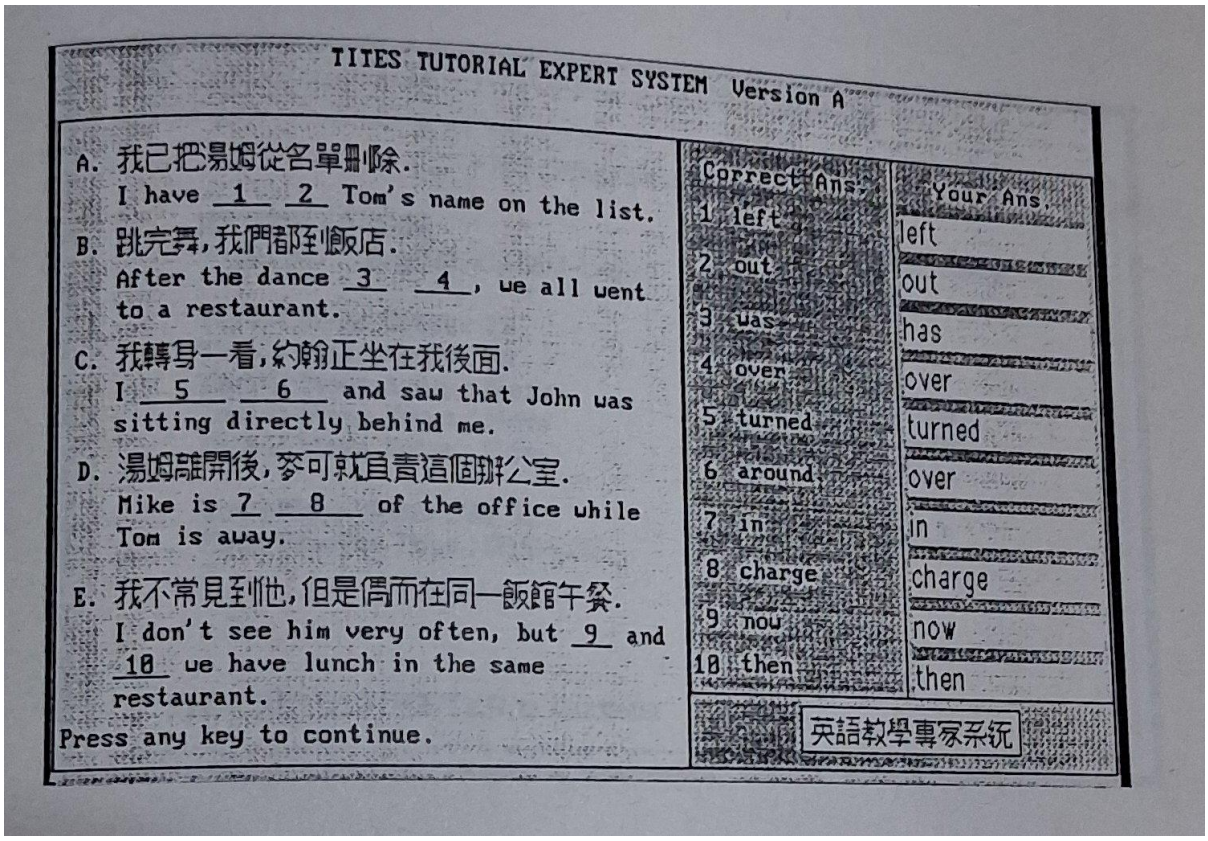


Figure 4-13. Questions, user's answers, and correct answers for Test-C.

All five types of tests share similar procedures so the user can manipulate the tests in the same way. After users have completed the instructions and examinations, the system will produce a final score report (as shown in Figure 4-15) and update all the records in dBASE III data files. At this time, users can press the F2 key to exit or restart the system.

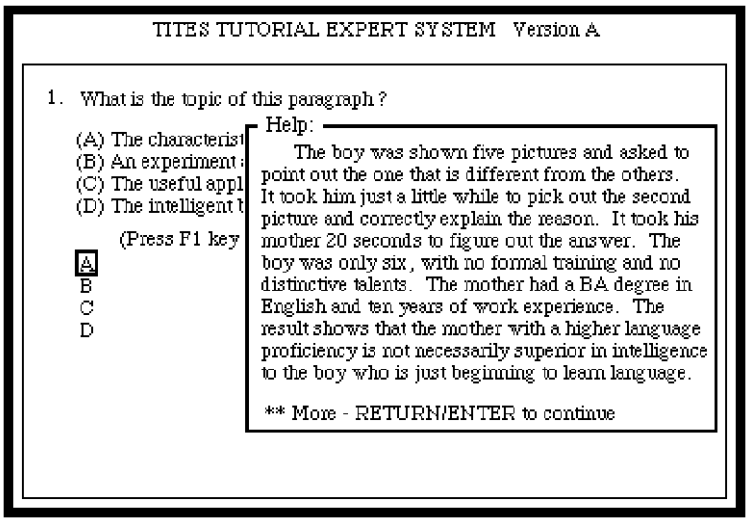

TITES TUTORIAL EXPERT SYSTEM Version A
1. What is the topic of this paragraph ?
(A) The characterist
(B) An experiment
(C) The useful appl
(D) The intelligent
(Press F1 key
A
B
C
D
Help:
The boy was shown five pictures and asked to point out the one that is different from the others. It took him just a little while to pick out the second picture and correctly explain the reason. It took his mother 20 seconds to figure out the answer. The boy was only six, with no formal training and no distinctive talents. The mother had a BA degree in English and ten years of work experience. The result shows that the mother with a higher language proficiency is not necessarily superior in intelligence to the boy who is just beginning to learn language.
** More - RETURN/ENTER to continue

Figure 4-14. Reading comprehension test of Test-E.

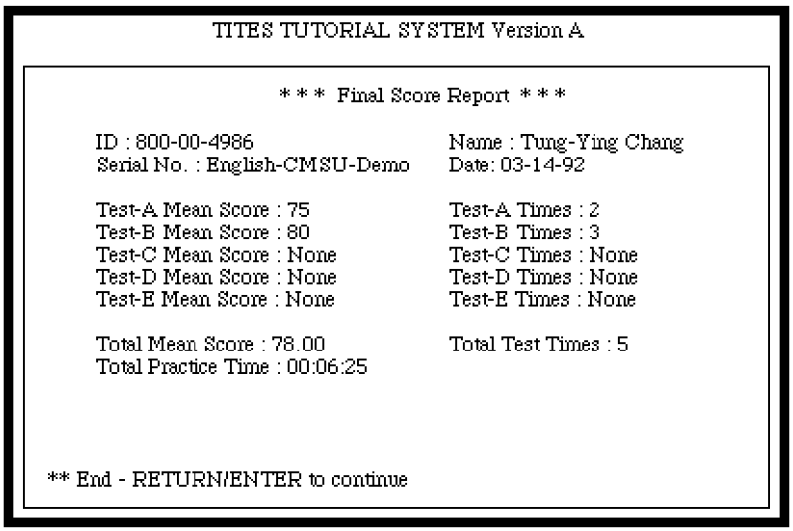
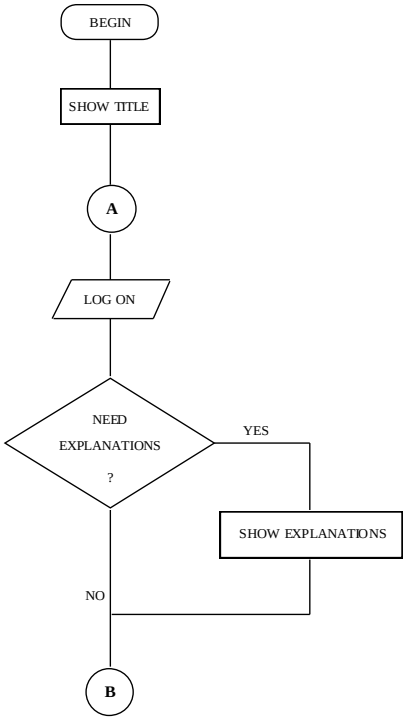

TITES TUTORIAL SYSTEM Version A
*** Final Score Report ***
ID : 800-00-4986
Name : Tung-Ying Chang
Serial No. : English-CMSU-Demo
Date: 03-14-92
Test-A Mean Score : 75
Test-A Times : 2
Test-B Mean Score : 80
Test-B Times : 3
Test-C Mean Score : None
Test-C Times : None
Test-D Mean Score : None
Test-D Times : None
Test-E Mean Score : None
Test-E Times : None
Total Mean Score : 78.00
Total Test Times : 5
Total Practice Time : 00:06:25
** End - RETURN/ENTER to continue

Figure 4-15. Final score report of the examination.

Operating Flow Diagram

TITES was developed with highly interactive and intelligent use of graphics for input and output. All the data were manipulated by a set of routines with online help facilities. Figure 4-16 shows a flow diagram for typical operation of TITES.



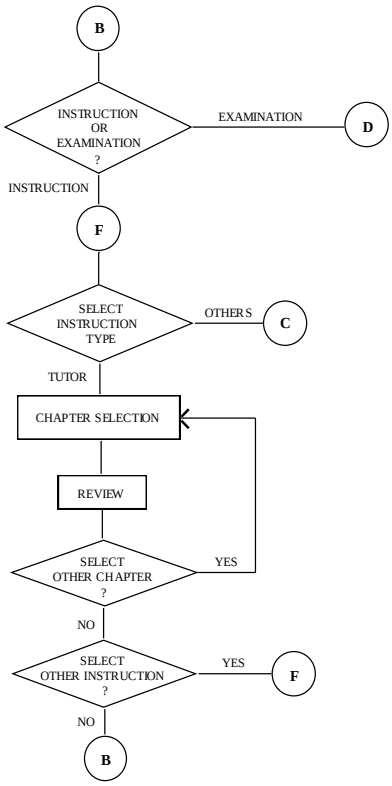
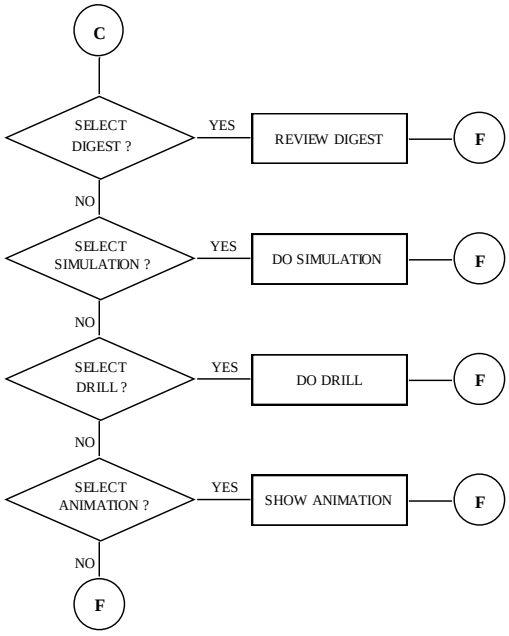


Figure 4-16. Typical operating flow diagram of TITES (continued)



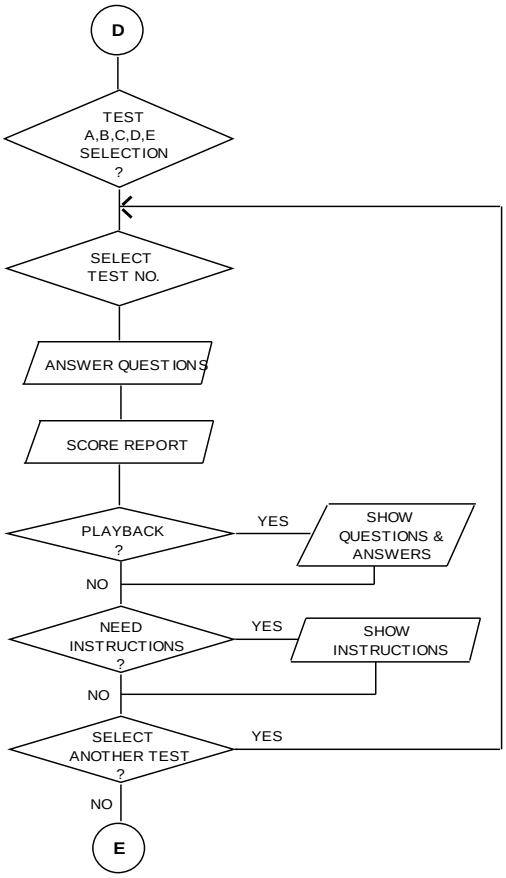


Figure 4-16. Typical operating flow diagram of TITES (continued)

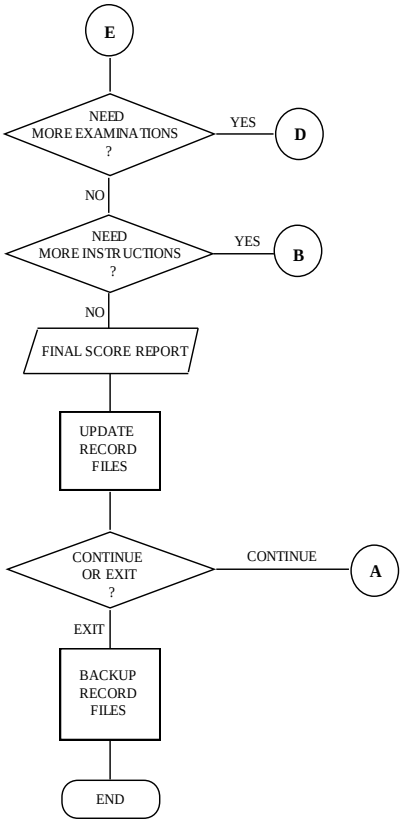


Figure 4-16. Typical operating flow diagram of TITES.

General Architecture of the System

In PC Plus, the knowledge base was constructed in hierarchical frame structures. TITES can be divided into three major frame-tree structures. The first frame, a set of the CAI family, consists of several sub-frames such as drill, simulation, and tutor. The tutorial materials could be textual, graphic, or animated. The second frame, a set of the CMI family, is also a frame-tree structure. The CMI frame is responsible for presenting problems or diagnostic information and recording the student's response to update the student database. The third frame is the text file editor which allows the author to generate teaching material, test problems, and answers. Figure 4-17 shows the overall frame-tree architecture of the system. Each square is a frame in which goals, parameters, and rules are provided and, in some way, can be regarded as an independent knowledge base.

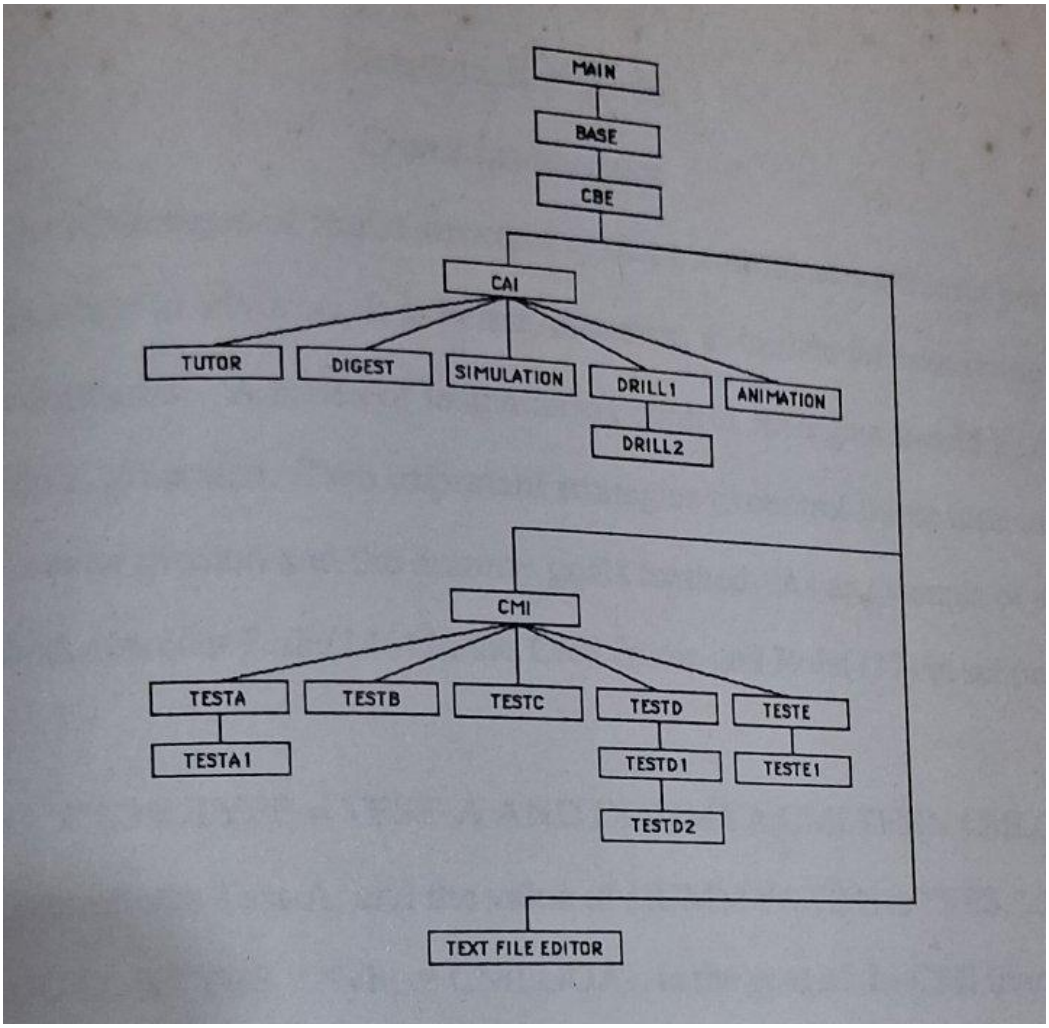


Figure 4-17. Overall frame-tree architecture of TITES.

The reasons to apply frame structure are:

1. To divide the knowledge base into several small subject-related hierarchical knowledge bases. This kind of organizational skill simplifies the complicated structure and makes problems logical and clear.

2. A sub-frame can be instantiated repeatedly at any time. This allows for variation in program control from a given point and switches to other parts or repetitions of particular items and fulfills the "branching" requirement of CAI.

3. By means of frame instantiated procedure and inference rules, an inference engine can distinguish a user's level of sophistication and make some adjustment to suit many needs of individual users, such as test-type, level of difficulty, progress of instruction, and frequency of review. The benefit of learner control, therefore, can be reached in both flexibility and efficiency.

Programming Tactics

Frame Instantiation

One of the advantages of frame structure is that hierarchical subframes permit instantiating repeatedly at anytime. It is better, however, to initiate the root frame before subframes are constructed. A series of instantiating control strategies should be considered before the system is generated. Two important strategies to control frame instantiating are the dummy parameter method and the dummy goals method. As an example of the dummy parameter method, consider Rule(146) in the CMI frame and Rule(117) in subframe TESTA frame.

Rule146: If CMI.TYPE = TEST-A AND DUMMYA.CMI THEN CMI.GOAL

(If the user selects Test-A, and the value of DUMMYA.CMI is "YES," then the value of CMI.GOAL is "YES." Where CMI.GOAL is the goal of the CMI frame, CMI.TYPE and DUMMYA.CMI are parameters of the CMI frame.)

Rule117: IF CMI.TYPE = TEST-A THEN DUMMYA.CMI

(If the user selects Test-A, then the value of DUMMYA.CMI is "YES.")

In this case, DUMMYA.CMI is a check parameter or a dummy parameter. The purpose of using a dummy parameter is to ensure that the inference engine checks the logic in the specified subframe. At the beginning of normal backward chaining, the system searches the value of goal parameter CMI.GOAL by finding Rule(146). Because the user selects TEST-A, the value of CMI.TYPE was set. In order to find the value of DUMMYA.CMI, the system must try Rule(117) in TESTA frame. By placing a dummy parameter in the ancestor frame (CMI) and ordering appropriate rules, the system will instantiate the specified subframe (TESTA).

Another example is the dummy goal method used in ancestor frame CAI and subframe TUTOR. Where DUMMYT.GOAL is a dummy goal in the CAI frame, FUNCTION is a parameter of the TUTOR frame. To instantiate subframe TUTOR, a Rule(045) which sets the dummy goal's value was organized in the TUTOR frame.

Rule045: IF FUNCTION = TUTOR THEN DUMMYT.GOAL

(If the user selects Tutor instruction, then the value of DUMMYT.GOAL is "YES.")

In this case, DUMMY.GOAL is not a real goal of the CMI frame, the only purpose for using DUMMYT.GOAL is to force the system to instantiate the TUTOR frame.

In the construction of TITES's knowledge base, both the dummy parameter method and dummy goal method were used to control subframe instantiation. For example, during the process of multi-goals inference, the use of these two control strategies cooperated with a complete tracing in backward chaining can force the system to trace the dummy goal and infer the value of the real goals one by one. This inference method named "Hungry Tracing" given flexibility in causing frame instantiation, and it can be considered as an efficient design strategy of rule-based systems. In TITES, more examples reveal that the hungry tracing method can get expected results even in the hybrid inference of forward chaining and backward chaining.

External Program Interface

In TITES, teaching materials were separated from the tutorial system by means of several graphic routines and external program interfaces. The system instantiates test frames to dump testing programs to a data buffer. When the user decides to use the tutor or test module, each parameter reads individual programs and answers from the buffer. Teaching and testing materials were designed in ASCII files or graphics with an identical serial number. The courseware can be stored on the disk or CD-ROM. The student's records and responses will be updated and kept in the database through external interfaces which can communicate with DOS, dBASE III, LOTUS 123, or other software. The designs not only isolate courseware from the tutorial system but also allow the author to make changes for efficient and flexible lesson design.

Chinese Characters and Graphics Processing

Because the Chinese language is based upon thousands of different characters, a computer that can apply Chinese characters must be equipped with a powerful CPU and flexibility for graphics generation and processing. Several kinds of Chinese input systems have been developed in the past five years. Though software compatibility and character input have hampered Chinese CAI, the development of hardware supports flexible, high resolution graphics which seem to offer a solution to the problem.

There are two different methods of initiating graphic displays in IBM PC compatible computers. One is bit-mapping in which images consist of pixels. The other is object-oriented graphics in which images are created by lines or other geometric patterns. For example, in EGA display, to draw a line from point A to point B, bit-mapped graphics light every pixel between point A and point B; object-oriented graphics apply an algorithm to link A and B. Each method has advantages. Bit-mapped images can display subtle and irregular shapes; object-oriented graphics can generate smooth patterns and require less disk space. In the speed of display, however, bit-mapped images are faster than object-oriented graphics. Based on the considerations of display speed and media space, it is obvious that bit-mapping is better for Chinese characters display.

Most expert system shells take a large amount of memory in order to allow the full power of AI based reasoning techniques to be applied to them. The Chinese input system also requires several hundred kilobytes of memory to be used successfully. In addition to memory requirements, there are some interference problems between the expert system shell and the Chinese input systems. In TITES, the design strategy is to transfer Chinese characters to an image and simplify these problems.

Numerous programming languages can be used to edit and display Chinese images together with English texts or graphics. Sophisticated word processors can do similar work. In order to grab Chinese characters, a set of programs were designed in which Chinese images can be transferred to Paint Brush PLUS, Dr. Halo III, AUTOCAD, or Lotus 123. Such an approach is far less flexible than using the Chinese input system. But it increases the efficiency of the computer's processing and prevents interferences. This is because the computer no longer has to reserve all Chinese characters and the Chinese input system is not stored in memory. After finishing graphics transformation, the author can rotate, revise, stretch, enlarge, and position Chinese images within the graphic software environment. The integration of images from various support tools successfully solves the problem of interferences between MS-DOS and the Chinese input system.

Text File Editor

Many word processing programs have been marketed during the last decade. Some of them are sophisticated, powerful, and expensive. Although they can be used to create ASCII files, there is not an efficient tool to manage hundreds or thousands of ASCII files in current word processors. The text file editor designed for courseware generating was devised from the concept of combining data management and word processing techniques. Suppose each database is a book, then each record can be considered as a page in the book. In the text file editor, an ASCII text file will be generated and managed as a record in a database. The courseware author can construct a data structure and give a unique serial number to individual courseware. A built-in text editor, a random-access index routine, and a dynamic data structure make up the program to perform operations of appending, browsing, coping, deleting, and updating records. Other functions such as find, print, index generation, database information display, and data transfer can help the courseware author in editing work. Figure 4-18 shows the main menu of the text file editor. Using the file name as an index key, the courseware author can manipulate all the text files in a database identified by the serial number or specified title.

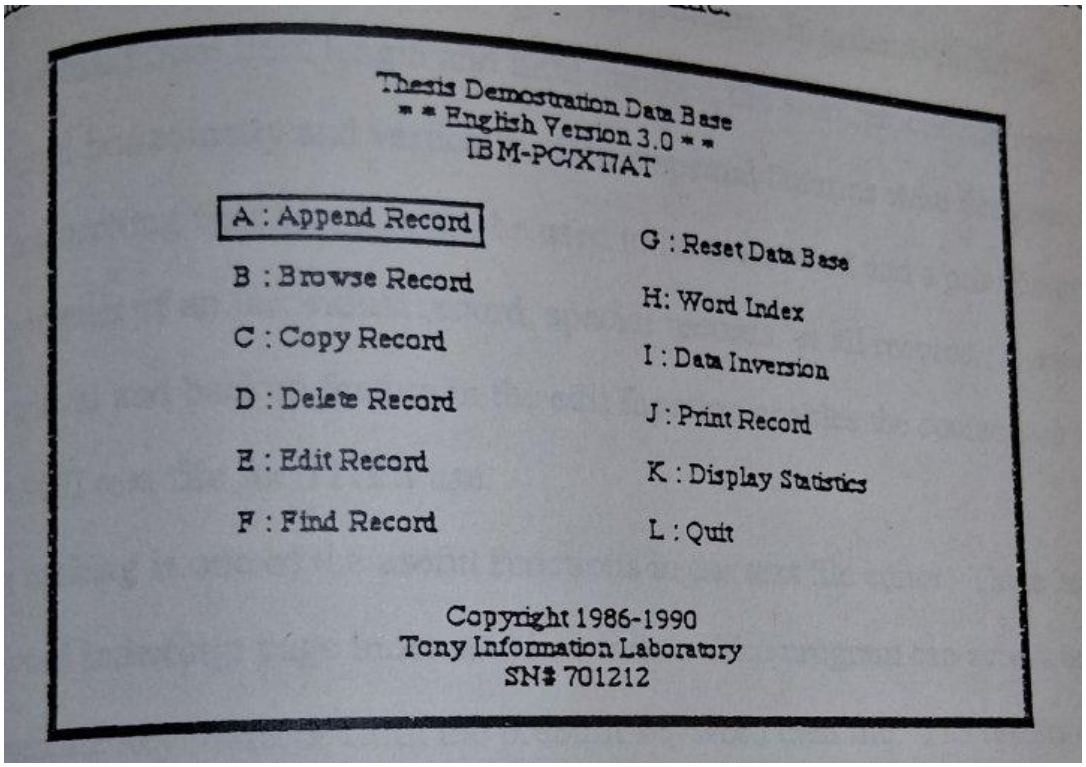


Figure 4-18. Main menu of the Text File Editor.

The text file editor is a conventional program written in Microsoft's Quick BASIC language. With key file random access, the user can retrieve the desired file and directly read or write at any location on a database. A data file, an index file, a data structure file, and an autoexec file make up the file structure of the text file editor. The data file is a random file containing the text materials entered by the user. The index file is a binary file containing pointers to the data file so that the user can access the record with a key value such as the ASCII text file name. The data structure file is a sequential file containing the necessary data such as field number, field name, field length, and record capacity, which enable the program to construct a database. The autoexec file provides the names of files that make up the database and other information which enables the courseware author to access the individual database whenever needed.

As Quick BASIC supports dynamic arrays, the maximum capacity in a database can be expanded to 10,000 records. The maximum number of fields in a record is 201; the maximum field length is 142 characters. Theoretically, there is no limitation to the number of databases except that imposed by the disk capacity. The built-in word editor is a subprogram co-operated with the data management routines. In order to fulfill the requirements of maximum field length and field number, the word processing routine was designed to scroll horizontally and vertically. Some special features were designed in the word editor: a searching function that can be used to locate a word and a print function to print partial contents of an individual record, special records, or all records. A portable ASCII file retrieval and backup feature in the edit function enables the courseware author to create final ASCII text file for TITES use.

Index making is one of the useful functions in the text file editor. There are two types of keyword indexing: page index and line index. The program can accept keyword data either from the keyboard or from the prebuilt keyword data file. The results of keyword indexing will be saved in an ASCII file which includes the name of the database, keyword source file, and word index list. A password routine was attached to the editor which enables the authorized user access to the editor and prevents the courseware from unexpected modification. The text file editor is also bilingual; it can be used to create

English, Chinese, or Chinese-English hybrid text files. Working with a resident graphic capture program, the text file editor can be used as a Chinese graphic generator for editing Chinese tutorial and test materials.

In addition, the text file editor provides a temporary exit function which allows the courseware author to invoke MSDOS during modification. Through the main menu of the text file editor, the tutorial system can be terminated by pressing the Esc key. The courseware author can run other programs at the DOS prompt or type "EXIT" to return to the text file editor to continue.

CHAPTER V

CONCLUSIONS AND RECOMMENDATIONS

The woods are lovely, dark, and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.

Robert Frost

Conclusions

One of the goals of the study was to create a programming-free administration and instruction environment to enable teachers to generate low-cost multimedia-based bilingual courseware. Based on that perspective, the approach focused on pragmatic issues rather than on theoretical subjects. With a basic understanding of computer programming, both expert systems techniques and conventional programming were used to construct the system.

TITES, a prototype of an tutorial expert system, was developed earlier by the researcher. The system can be used as a personal research tool; it also offers the researcher a pregnant experimental environment for programming. Some concepts and ideas were implemented in the system, such as course-independent design, the use of graphics to present Chinese characters, combining database management and word processing techniques to create a text file editor, and the use of the hungry-tracing method in the hybrid inference of forward chaining and backward chaining.

During the development process, it was confirmed that developing an expert system without conventional programming was not efficient. Although expert system development shells provide specific techniques for handling knowledge representations and inference mechanisms, its vast memory requirements and complicated inference processes frequently results in garbage collection that slows down the system's executing rate. In other words, the flexibility of the expert system is obtained at the cost of its efficiency. On

the contrary, conventional programming techniques are more efficient but less flexible.

The study also revealed that developing an expert system requires long-term efforts and a wide variety of background knowledge; hence, it is time-consuming and expensive. Figure 5-1 shows the development time of each implementation item.

Implementation Items	Time (hours)
Pilot Systems Design	380
Shell Study & Knowledge Base Design	1120
Text File Editor Design	960
Auxiliary Programs Design	160
Testing & Debugging	600
Total Time Used	3220

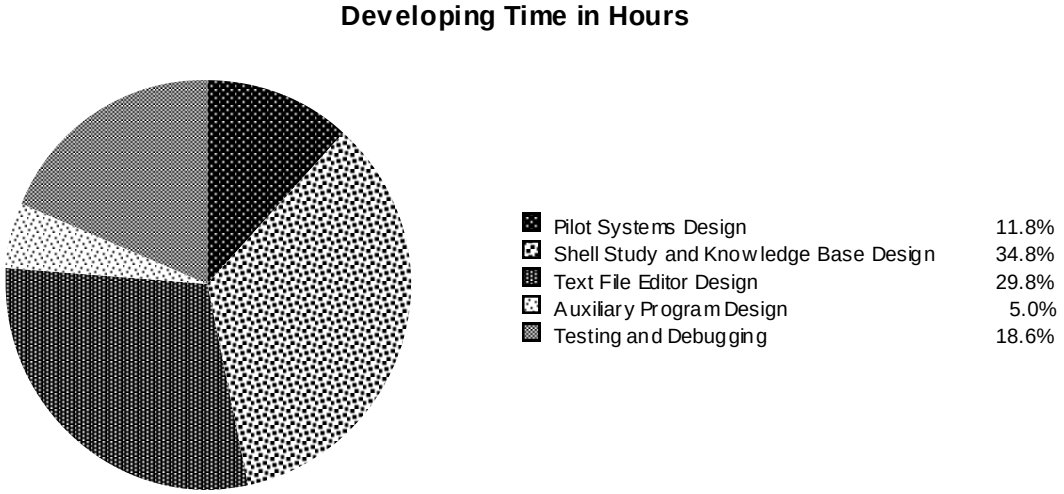

Developing Time in Hours
Pilot Systems Design 11.8%
Shell Study and Knowledge Base Design 34.8%
Text File Editor Design 29.8%
Auxiliary Program Design 5.0%
Testing and Debugging 18.6%

Figure 5-1. Development Time in Hours: Where the Time Went.

Tutorial expert systems can be regarded as a combination of art, education, and scientific technologies. As more and more information and technologies become available with the use of computers, the process of learning will change. As people become more knowledgeable, they want to know more and they want more indepth information. Under

the influence of the proliferation of microcomputers and the increasing costs of traditional education, using personal computers as auxiliary teaching and learning tools is not only a trend but a fact that educators must recognize. Unless educators keep track of the efforts made by modern scientists and keep abreast of technological change, they will lose the battle as effective instructors.

The design philosophy of TITES is to create an experimental environment for one of the knowledge navigators. Turing (1950) said, "We can see only a short distance ahead, but we can see plenty there that needs to be done" (p. 35). Research is like opening the door; there is always another door inside the open door. Developing TITES not only verified the feasibility of applying expert system techniques in building educational software, it also holds the key to another door for future research.

Recommendations for Future Research

The system described above is intended to be an instructional tool for teaching consultants. It requires consideration of teaching material and student involvement in the form of an experimental tutor and the provision of information. There are still several limitations in the current version of TITES. Some students may prefer a more active learning style and feel excited if they can ask some questions and communicate with the computer in natural language. As current linguistic and AI research have made some breakthrough in natural language understanding, it is believed that intelligent dialogue functions will become possible in the near future.

Another problem stems from the limited feedback available. Although feedback is one of the most important functions in the expert system, the assessment of the student's learning is still the weakness of the study. Unlike a physics or chemical laboratory experiment, it is very difficult for the system to apply quality or quantity analysis of student performance. Individual differences among students, such as learning curves, background knowledge, and personality, make feedback functions more difficult.

Applying a student-learning model to install a more efficient feedback function will be feasible; however, it involves a relatively large student sample. The tutorial system, equipped with a highly flexible teaching model which can survey students' learning curves,

their potential and achievement, can provide the right teaching method for each student. From this point of view, machine learning could be the way to a more flexible teaching model.

Another way to make the tutorial system more intelligent is to develop intelligent-design courseware. One of the important responsibilities of the teacher is to orchestrate student learning. Designing courseware is not just to place lessons or tests in files or graphics. The courseware author must set up the objectives for the lessons, write a behavioral objective for each concept to be taught, and decide how to measure whether the objectives have been met. In other words, the author should make a logical analysis of the courseware. This analysis includes applications of different domains: cognitive psychology, subject knowledge, instructional technology, Boolean analysis, syllogisms, and synthesis. Once the courseware has been analyzed precisely and logically, effective feedback can be provided to direct the user toward the correct learning modes.

Applying audio and video techniques to strengthen students' learning will be a valuable experiment for the tutorial system. Sound is the sensation that is produced when auditory nerves are stimulated by vibrating air molecules. It is an analog format signal. To reproduce or simulate a sound effect on a computer, it is necessary to employ digitizing techniques. Digitizing converts a sound from analog to digital format (see Figure 5-2).

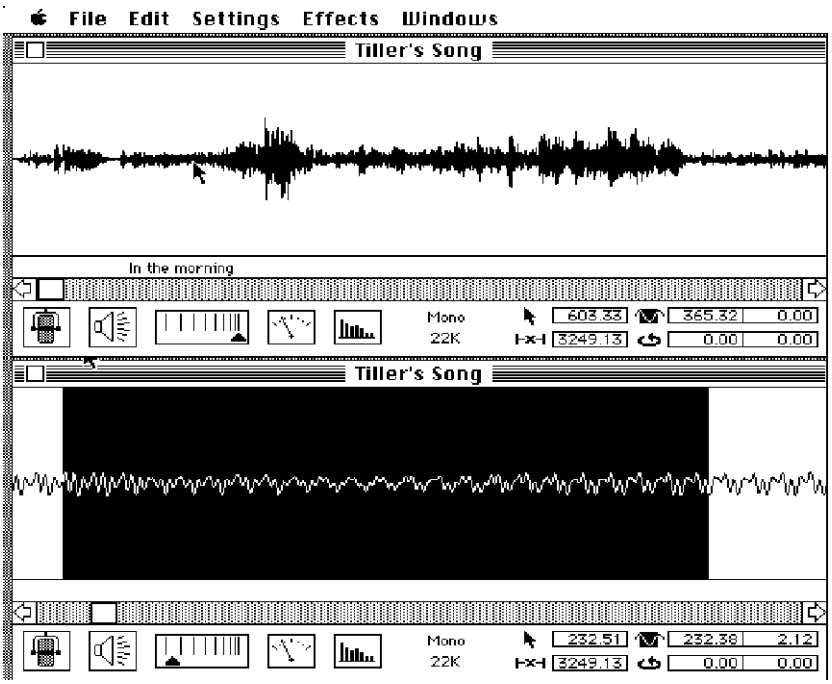


Figure 5-2. Digital sound format of a Taiwanese folk song presented in the MacRecorder.

Image processing is another changing technique. The 256 colors available with VGA cards on today's personal computers cannot present high quality graphics. The 32-bit color or "true color" technique can provide 16.7 million colors to display perfect images. The computer, however, requires much more data storage space and takes more time to display. A more practical method of adding video effects to the tutorial system is to design an external video interface for the videodisc player. A Constant Angular Velocity (CAV) videodisc can contain up to 54,000 frames of addressable video images. Using a "search" command, users can move through the video frames sequentially or randomly. By means of programming control, "Play" and multispeed commands display motion sequences at normal (30 frames per second), slow, or fast speed in forward or reverse.

It can be expected that mass storage techniques and graphics coprocessors will be improved in the near future. As a low-cost desktop video production system and the re-writable CD-ROM SCSI disk drive are marketed, it will become possible to integrate more and more multimedia technologies into tutorial expert systems.

Expert systems provide a good experimental environment for intelligent tutorial modules. It is possible to build an intelligent tutorial module based on meta-level knowledge in rule-based systems. Any system that lacks feedback and automatic learning

features cannot be a “true intelligent” system. One of the next goals of TITES is to integrate CAI, database, expert system, and multimedia technologies to construct a knowledge base, with the ability to exhibit behavior classified as "an intelligent tutor." The development of TITES is a definite effort in this direction. The final goal of TITES is to be an artificial tutor expert that has the capability to teach, to communicate with the student, to know what to teach, and to be an assistant to the teacher.

REFERENCES

Alessi, S. M., & Shih, Y. -F. (1989). The growth of computer-assisted instruction in Taiwan schools. *Computers Education, 13*(4), 337-341.

Ambron, S., & Hooper, K. (1988). *Interactive multimedia: Visions of multimedia for developers, educators and information provider*. Redwood, WA: Microsoft Press.

Barr, A., & Feigenbaum, A. (1981). *The handbook of artificial intelligence* (Vols. I-III). Standford, CA: HeurisTech.

Bielawski, L., & Lewand, R. (1988). *Expert systems development: Building PC-based applications*. Wellesley, MA: QED Information Sciences.

Bielawski, L., & Lewand, R. (1991). *Intelligent systems design: Integrating expert systems, hypermedia, and database technologies*. New York: John Wiley & Sons.

Bitzer, D. L. (1986). The PLATO project at the University of Illinois. *Engineering Educ*ation, *77*(3), 175-180.

Boardman, A.D., Cooper, B. W. J., Keeler, G. J., & Swage, J. (1988). Software development for undergraduates in physics. *Computers Education, 12*(1), 29-35.

Bush, V. (1945, July). As we may think. *Atlantic Monthly*, pp. 101-108.

Chang, Tung-Ying (1985, September). Natural language study and automatic translation system. [Document in Chinese, title translated.] *Autotech Magazine*, pp. 73-82.

Chang, Tung-Ying (1987a, July). The veil of intelligence. [Document in Chinese, title translated.] *Autotech Magazine*, pp. 97-102.

Chang, Tung-Ying (1987b, December). Semantic network and frame structure. [Document in Chinese, title translated.] *Autotech Magazine*, pp. 211-221.

Chang, Tung-Ying (1991, October). Interactive multimedia. [Document in Chinese, title translated.] *Autotech Magazine*, pp. 111-116.

Chase, S. (1954). *Power of words*. New York: Harcourt, Brace.

Chen, Ching-chih (1989). *HyperSource on multimedia/hyperMedia technologies.* Chicago, IL: American Library Association.

Chomsky, N. (1980). *Rules and representations*. New York: Columbia University Press.

Ciser, S. (1990). Visual almanac: Interactive multimedia on the Mac. *Online*, *14*(2), 87-90.

Clancey, W. J. (1987). *Knowledge based tutoring: The GUIDON program*. Cambridge, MA: MIT Press.

Dever, S. Y., & Pennington, M. C. (1989). Computer capabilities underlying computer-learner interaction. In M. C. Pennington (Ed.), *Teaching languages with computers: The state of the art* (pp. 11-28). La Jolla, CA: Athelstan.

D'Ignazio, F. (1989). *Scholastic guide to classroom multimedia*. New York: Scholastic.

Duchastel, P. (1989). ICAI systems: Issues in computer tutoring. *Computers Education*, *13*(1), 95-100.

Duda, R. O. (1979). Model design in the PROSPECTOR consultant system for mineral exploration. In D. Michie (Ed.), *Expert systems in the microelectronics age* (pp. 153-167). Edinburgh, Scotland: Edinburgh University.

Elsom-Cook, M.T., & O'Malley, C. E. (1990). ECAL: Bridging the gap between CAL and intelligent tutoring systems. *Computers Education*, *15*(1-3), 69-81.

Feigenbaum, E., Buchanan, B., & Lederberg, J. (1971). On generality and problem solving: A case study using the DENDRAL program. In B. Meltzer & D. Michie (Eds.), *Machine Intelligence 6* (pp. 165-190). New York: American Elsevier.

Fersko-Weiss, H. (1985). Expert systems: Decision-making power. *Personal Computing*, *9*(11), 97-105.

Fletcher, J. D. (1990). *Effectiveness and cost of interactive videodisc instruction in defence training and education* (Report No. IDA Paper P-2372). Alexandria, Virginia: Institute for Defense Analyses.

Freedman, R. S., & Rosenking, J. P. (1986). Designing computer-based training systems: OBIE-1:KNOBE. *IEEE Expert*, *1*, 31-38.

Gratch, B. (1986). Computer-assisted instruction: An unfulfilled promise. *Wilson Library Bulletin*, *61*(4), 20-22.

Grigonis, R. W. (1987, April). MYCIN-like expert systems. *Dr. Dobb's Journal*, pp. 42-82.

Hayes-Roth, F., Waterman, D.A., & Lenat, D.B. (1983). *Building expert systems*. Reading, MA: Addison-Wesley.

Heid, G. (1991). A critical comparison of the Macintosh and IBM PC worlds. *MacWorld*, *8*(3), 120-129.

Johnson, B. L., Bergeron, R. D., & Malcolm, P. (1990). Modeling the teaching consultant. *Computers Education*, *14*(2), 125-136.

Jones, C., & Fortescue, S. (1987). *Using computers in the language classroom*. New York: Longman.

Lawson, V. L. (1988). Using a computer-assisted instruction programs as an alternative to the traditional library orientation / instruction tour: An evaluative study (Doctoral dissertation, The Florida State University, 1988).

Lemonick, M. (1984). Machines with living parts. *Science Digest*, *92*(2), 26.

Lu, S. C-Y. (1989, January). *Artificial intelligence techniques for engineering automation*. Unpublished lecture notes. National Cheng Kung University.

Matta, K. F., & Kern, G. M. (1989). A framework for research in computer-aided instruction: Challenges and opportunities. *Computers Education*, *13*(1), 77-84.

McDermott, J. (1982). R1: A rule-based configurer of computer systems. *Artificial Intelligence*, *19* (1), 39-88.

Melle, W., Shortliffe, E. H., & Buchanan, B. G. (1984). *EMYCIN: A knowledge engineer's tool for constructing rule-based expert systems*. New York: Addison-Wesley.

Miller, M. J. (1989). Multimedia technology is not just the buzzword of the year. *InfoWorld*, *11*(15), 56.

Minsky, M. (1975). A framework for representing knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision* (pp. 211-277). New York: McGraw-Hill.

Newell, A., Shaw, J. C. & Simon, H. A. (1957). Empirical explorations of the logical theory machine. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 109-133). New York: McGraw-Hill.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Newhard, R. (1987). Converting information into knowledge: The promise of CD-ROM. *Wilson Library Bulletin*, *62*(4), 38.

Nyns, R. R. (1990). An expert system in computer assisted language learning. *Computers Education* , *15*(1-3), 99-103.

Palmer, F. R. (1981). *Semantics*. New York: Cambridge University Press.

Papert, S. (1970). Teaching children thinking. In R. P. Taylor (Ed.), *The computer in the school: Tutor, tool, tutee* (pp. 161-176). New York: Teachers College Press.

Piaget, J. (1970). *Structuralism*. New York: Basic Books.

Rambally, G. K. (1986). The AI approach to CAI. *The Computing Teacher*, *137*, 39-42.

Reasor, E. (1985). *The expert system inference engine* [Computer program manual]. Tampa, FL: Lightwave Consultants.

Schank, R. C. (1972). Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, *3*(4), 552-631.

Schank, R. C. (1984). *The cognitive computer*. Reading, MA: Addison-Wesley.

Shortliffe, E. H. (1976). *Computer-based medical consultation: MYCIN*. New York: American Elsevier.

Slobin, D. I. (1979). *Psycholinguistics.* Glenview, IL: Scott, Foresman.

Texas Instruments. (1987). *Personal consultant plus* [Computer program manual]. Austin, TX: Texas Instruments, Data System Group.

Turing, A. (1950). Computing machinery and intelligence. In E. A. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 11-35). New York: McGraw-Hill.

Walsten, D. (1991). NCSA video Macintosh: Producing scientific visualizations on a desktop. *Data Link*, *5*(4), 22-26.

Wu, T. -H. (1987). CAI in Taiwan: State and problems. *Journal of Computer-Based Instruction*, *14(*3), 104-106.

CERTIFICATE OF REGISTRATION

This Certificate issued under the seal of the Copyright Office in accordance with title 17, United States Code, attests that registration has been made for the work identified below. The information on this certificate has been made a part of the Copyright Office records.

REGISTER OF COPYRIGHTS
United States of America

FORM TX
UNITED STATES COPYRIGHT OFFICE

REGISTRATION NUMBER
TX 3 400 219

EFFECTIVE DATE OF REGISTRATION
June 24, 1992
Month Day Year

DO NOT WRITE ABOVE THIS LINE. IF YOU NEED MORE SPACE, USE A SEPARATE CONTINUATION SHEET.

1 TITLE OF THIS WORK ▼
A MULTIMEDIA-BASED BILINGUAL INSTRUCTIONAL SYSTEM USING AN EXPERT SYSTEM SHELL

PREVIOUS OR ALTERNATIVE TITLES ▼

PUBLICATION AS A CONTRIBUTION If this work was published as a contribution to a periodical, serial, or collection, give information about the collective work in which the contribution appeared. Title of Collective Work ▼

If published in a periodical or serial give: Volume ▼ Number ▼ Issue Date ▼ On Pages ▼

2 a
NAME OF AUTHOR ▼
TUNG-YING CHANG
DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼
1950
Was this contribution to the work a "work made for hire"? ☐ Yes ☒ No
AUTHOR'S NATIONALITY OR DOMICILE Name of Country OR Citizen of ▶ TAIWAN Domiciled in ▶
WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☒ No Pseudonymous? ☐ Yes ☒ No If the answer to either of these questions is "Yes," see detailed instructions.
NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼
ENTIRE TEXT AND FIGURES

NOTE Under the law, the "author" of a

b
NAME OF AUTHOR ▼
DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼
Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No
AUTHOR'S NATIONALITY OR DOMICILE Name of Country OR Citizen of ▶ Domiciled in ▶
WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☐ No Pseudonymous? ☐ Yes ☐ No If the answer to either of these questions is "Yes," see detailed instructions.
NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

c
NAME OF AUTHOR ▼
DATES OF BIRTH AND DEATH Year Born ▼ Year Died ▼
Was this contribution to the work a "work made for hire"? ☐ Yes ☐ No
AUTHOR'S NATIONALITY OR DOMICILE Name of Country OR Citizen of ▶ Domiciled in ▶
WAS THIS AUTHOR'S CONTRIBUTION TO THE WORK Anonymous? ☐ Yes ☐ No Pseudonymous? ☐ Yes ☐ No If the answer to either of these questions is "Yes," see detailed instructions.
NATURE OF AUTHORSHIP Briefly describe nature of the material created by this author in which copyright is claimed. ▼

0493409728

3 a YEAR IN WHICH CREATION OF THIS WORK WAS COMPLETED This information must be given in all cases.
1992 ◀ Year

b DATE AND NATION OF FIRST PUBLICATION OF THIS PARTICULAR WORK Complete this information ONLY if this work has been published.
Month ▶ MAY Day ▶ 10 Year ▶ 1992
U.S.A ◀ Nation

4 COPYRIGHT CLAIMANT(S) Name and address must be given even if the claimant is the same as the author given in space 2. ▼
TUNG-YING CHANG
P.O. BOX 530
TAINAN, TAIWAN

See instructions before completing this space

TRANSFER If the claimant(s) named here in space 4 are different from the author(s) named in space 2, give a brief statement of how the claimant(s) obtained ownership of the copyright. ▼

DO NOT WRITE HERE OFFICE USE ONLY
APPLICATION RECEIVED
JUN. 24. 1992
ONE DEPOSIT RECEIVED
TWO DEPOSITS RECEIVED
JUN. 24. 1992
REMITTANCE NUMBER AND DATE

DO NOT WRITE HERE

MORE ON BACK ▶ • Complete all applicable spaces (numbers 5-9) on the reverse side of this page. • See detailed instructions.